

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: KHOA HỌC MÁY TÍNH

**ĐỀ TÀI: XÂY DỰNG PHẦN MỀM HỖ TRỢ PHÂN TÍCH
DỮ LIỆU VÀ DỰ BÁO**

Giảng viên hướng dẫn: TS. Nguyễn Mạnh Cường

Lớp: KHMT01 – K15

Sinh viên thực hiện: Hà Tuấn Anh

Mã sinh viên: 2020607487

Hà Nội, 2024

MỤC LỤC

MỤC LỤC.....	1
MỤC LỤC HÌNH ẢNH	3
MỤC LỤC BẢNG BIỂU	6
LỜI CẢM ƠN	7
LỜI MỞ ĐẦU	8
CHƯƠNG 1 TỔNG QUAN VỀ PHÂN TÍCH DỮ LIỆU.....	10
1.1 Tổng quan về trí tuệ nhân tạo	10
1.1.1 Trí tuệ nhân tạo là gì.....	10
1.1.2 Các loại trí tuệ nhân tạo.....	10
1.1.3 Lợi ích và mặt trái của trí tuệ nhân tạo.....	12
1.1.4 Các ứng dụng của trí tuệ nhân tạo	14
1.2 Tổng quan về phân tích dữ liệu	16
1.2.1 Phân tích dữ liệu là gì.....	16
1.2.2 Quy trình phân tích dữ liệu.....	17
1.3 Tổng quan về bài toán dự báo	18
1.3.1 Lịch sử bài toán dự báo	18
1.3.2 Tình hình phát triển của bài toán dự báo ở Việt Nam	20
1.3.3 Tình hình phát triển của bài toán dự báo ở thế giới	21
CHƯƠNG 2 CÁC PHƯƠNG PHÁP PHÂN TÍCH MÔ TẢ VÀ DỰ BÁO	23
2.1 Giới thiệu tổng quan chương 2	23
2.2 Phương pháp phân tích mô tả	24
2.2.1 Phân tích mô tả	24
2.2.2 Phương pháp phân tích trên từng biến	25
2.2.3 Phương pháp phân tích trên nhiều biến	26
2.3 Phương pháp phân tích dự báo	27
2.3.1 Tổng quan về phân tích hồi quy.....	27
2.3.2 Các phương pháp phân tích hồi quy	28
2.3.3 Tổng quan về phân tích phân loại	30
2.3.4 Các phương pháp phân tích phân loại	30
2.4 Kết luận chương 2.....	43
CHƯƠNG 3 PHÂN TÍCH DỮ LIỆU BẰNG CÔNG CỤ PYTHON	44

3.1	Giới thiệu tổng quan chương 3	44
3.2	Phân tích mô tả bằng python	44
3.2.1	Phân tích thông kê mô tả bằng python	44
3.2.2	Trực quan hóa dữ liệu bằng python.....	48
3.3	Phân tích dự báo bằng python	52
CHƯƠNG 4	XÂY DỰNG PHẦN MỀM HỖ TRỢ PHÂN TÍCH DỮ LIỆU	58
4.1	Giới thiệu tổng quan chương 4.....	58
4.2	Giới thiệu framework sử dụng	58
4.3	Phân tích thiết kế hệ thống	61
4.3.1	Biểu đồ use case tổng quát	61
4.3.2	Mô tả chi tiết use case	61
4.3.3	Phân tích các use case	83
4.4	Thiết kế giao diện hệ thống	98
4.5	Các chức năng của hệ thống.....	101
KẾT LUẬN		115
TÀI LIỆU THAM KHẢO		116

MỤC LỤC HÌNH ẢNH

Hình 1.1: Quy trình phân tích dữ liệu	17
Hình 2.1: Sơ đồ tổng quát chương 2	23
Hình 2.2: Thuật toán máy hỗ trợ vec-tơ (SVM).	32
Hình 2.3: Thuật toán SVM tuyến tính.	34
Hình 2.4: Mô hình cây quyết định.	35
Hình 2.5: Hai phương pháp xác định độ thành công.	37
Hình 2.6: Hàm Entropy.	38
Hình 3.1: Sơ đồ tổng quát chương 3.	44
Hình 3.2: Dữ liệu về tập Stock.csv.	45
Hình 3.3: Tra cứu thông tin của tập dữ liệu.	46
Hình 3.4: Thống kê mô tả về tập dữ liệu.	47
Hình 3.5: Thống kê mô tả về tập dữ liệu bằng câu lệnh độc lập.	47
Hình 3.6: Trực quan hóa với biểu đồ đường.	48
Hình 3.7: Trực quan hóa với biểu đồ cột.	49
Hình 3.8: Trực quan hóa với biểu đồ hộp.	50
Hình 3.9: Trực quan hóa với biểu đồ scatter.	51
Hình 3.10: Trực quan hóa với biểu đồ heatmap.	52
Hình 3.11: Tiền xử lý dữ liệu trước khi chia.	53
Hình 3.12: Chia dữ liệu.	54
Hình 3.13: Chuẩn hóa dữ liệu.	54
Hình 3.14: Huấn luyện mô hình.	55
Hình 3.15: Đánh giá mô hình.	56
Hình 3.16: Dự đoán trên tập dữ liệu mới.	57
Hình 4.1: Sơ đồ tổng quát chương 4.	58
Hình 4.2: Logo streamlit.	59
Hình 4.3: Biểu đồ use case tổng quát.....	61
Hình 4.4: Biểu đồ trình tự use case tải dữ liệu lên.....	83
Hình 4.5: Biểu đồ lớp phân tích use case tải dữ liệu lên.	84

Hình 4.6: Biểu đồ trình tự xem hướng dẫn sử dụng	84
Hình 4.7: Use case xem hướng dẫn sử dụng phần mềm.....	85
Hình 4.8: Biểu đồ trình tự use case phân tích đơn biến.....	85
Hình 4.9: Biểu đồ lớp phân tích use case phân tích đơn biến.....	86
Hình 4.10: Biểu đồ trình tự use case phân tích đa biến.	86
Hình 4.11: Biểu đồ lớp phân tích use case phân tích đa biến.	87
Hình 4.12: Biểu đồ trình tự use case trực quan hóa dữ liệu.	87
Hình 4.13: Biểu đồ lớp phân tích use case trực quan hóa dữ liệu.	88
Hình 4.14: Biểu đồ trình tự use case phân tích dự báo.....	88
Hình 4.15: Biểu đồ lớp phân tích use case phân tích dự báo.....	89
Hình 4.16: Biểu đồ trình tự use case xử lý khuyết.....	89
Hình 4.17: Biểu đồ lớp phân tích use case xử lý khuyết.	90
Hình 4.18: Biểu đồ trình tự use case xử lý ngoại lai.....	90
Hình 4.19: Biểu đồ lớp phân tích use case xử lý ngoại lai.	91
Hình 4.20: Biểu đồ trình tự use case mã hóa dữ liệu.....	91
Hình 4.21: Biểu đồ lớp phân tích use case mã hóa dữ liệu.....	92
Hình 4.22: Biểu đồ trình tự use case nối dữ liệu.	92
Hình 4.23: Biểu đồ lớp phân tích use case nối dữ liệu.	93
Hình 4.24: Biểu đồ trình tự use case chọn tập X-Y.....	93
Hình 4.25: Biểu đồ lớp phân tích use case chọn tập X-Y.....	94
Hình 4.26: Biểu đồ trình tự use case chuẩn hóa dữ liệu.	94
Hình 4.27: Biểu đồ lớp phân tích use case chuẩn hóa dữ liệu.....	95
Hình 4.28: Biểu đồ trình tự use case chọn mô hình.....	95
Hình 4.29: Biểu đồ lớp phân tích use case chọn mô hình	96
Hình 4.30: Biểu đồ trình tự use case dự đoán dữ liệu.....	96
Hình 4.31: Biểu đồ lớp phân tích use case dự đoán dữ liệu	97
Hình 4.32: Biểu đồ trình tự use case lịch sử dự báo.....	97
Hình 4.33: Biểu đồ lớp phân tích use case lịch sử dự báo.....	98
Hình 4.34: Giao diện trang chủ của ứng dụng.....	98

Hình 4.35: Giao diện trang chủ của ứng dụng khí đã upload dữ liệu.....	99
Hình 4.36: Giao diện của tính năng phân tích mô tả.	99
Hình 4.37: Giao diện của tính năng phân tích dự báo.	100
Hình 4.38: Dữ liệu lịch sử dự báo.....	101
Hình 4.39: Chức năng upload dữ liệu	102
Hình 4.40: Source code optine menu.....	103
Hình 4.41: Chức năng upload dữ liệu.	103
Hình 4.42: Source code chức năng upload dữ liệu.....	104
Hình 4.43: Chức năng phân tích mô tả đơn biến.	105
Hình 4.44: Source code chức năng phân tích mô tả đơn biến.	105
Hình 4.45: Chức năng phân tích mô tả đa biến.	106
Hình 4.46: Source code chức năng phân tích mô tả đa biến.....	107
Hình 4.47: Chức năng trực quan hóa dữ liệu.....	108
Hình 4.48: Source code chức năng trực quan hóa dữ liệu.....	109
Hình 4.49: Chức năng tiền xử lý dữ liệu.....	110
Hình 4.50: Chức năng huấn luyện mô hình.	111
Hình 4.51: Source code chức năng huấn luyện mô hình.	111
Hình 4.52: Dự báo trên tập dữ liệu mới.	112
Hình 4.53: Lịch sử dự báo dữ liệu.	113
Hình 4.54: Dữ liệu sau khi được lưu về máy.....	114

MỤC LỤC BẢNG BIỂU

Bảng 2.1: Bảng mô tả dữ liệu	36
Bảng 4.1: Use Case tải dữ liệu lên	61
Bảng 4.2: use case xem hướng dẫn sử dụng phần mềm	63
Bảng 4.3: use case phân tích mô tả đơn biến	64
Bảng 4.4: use case phân tích mô tả đa biến	65
Bảng 4.5: use case trực quan hóa dữ liệu	67
Bảng 4.6: use case phân tích dự báo	68
Bảng 4.7: use case xử lý giá trị khuyết	69
Bảng 4.8: use case xử lý ngoại lai	71
Bảng 4.9: use case mã hóa dữ liệu	73
Bảng 4.10: use case nối dữ liệu	74
Bảng 4.11: use case chọn tập X và Y	76
Bảng 4.12: use case chuẩn hóa dữ liệu	77
Bảng 4.13: use case chọn mô hình	79
Bảng 4.14: use case dự đoán dữ liệu mới	80
Bảng 4.15: use case lịch sử dự báo	82

LỜI CẢM ƠN

Đầu tiên, em xin gửi lời cảm ơn đến quý thầy cô, nhà trường và bạn bè đã quan tâm và động viên chúng em trong suốt thời gian qua. Sự quan tâm và sự hỗ trợ của quý thầy cô không chỉ là nguồn động lực mà còn là một phần quan trọng trong việc xác định hướng đi và hoàn thiện đê tài của em.

Cá nhân em cũng muốn bày tỏ lòng biết ơn đặc biệt đến thầy TS. Nguyễn Mạnh Cường vì sự tận tâm và sự chỉ bảo tận tình trong quá trình hướng dẫn em. Những kiến thức mà thầy đã truyền đạt và những gợi ý quý báu đã giúp chúng em hiểu sâu hơn về phân tích và dự báo dữ liệu và áp dụng nó vào đê tài: “Xây dựng phần mềm hỗ trợ phân tích dữ liệu và dự báo”. Sự kiên nhẫn và sự đồng hành của thầy đã giúp em vượt qua những khó khăn và hoàn thành đê tài một cách tốt nhất.

Bản thân em cũng xin gửi lời cảm ơn đến tất cả những người đã giúp đỡ và hỗ trợ em trong quá trình nghiên cứu này. Những đóng góp, ý kiến và sự hỗ trợ của các bạn đã đóng vai trò quan trọng trong việc nâng cao chất lượng của báo cáo này.

Cuối cùng, em xin kết thúc lời cảm ơn này bằng việc cam kết sẽ tiếp tục nỗ lực, phấn đấu và trưởng thành hơn trong con đường nghiên cứu và học tập. Em sẽ luôn đặt lòng biết ơn và trân trọng những sự giúp đỡ mà em đã nhận được và sẽ cố gắng chia sẻ kiến thức và kinh nghiệm của mình để đóng góp vào sự phát triển của cộng đồng.

Sinh Viên

Hà Tuấn Anh

LỜI MỞ ĐẦU

Bước vào thời kỳ đỉnh cao của công nghệ, trí tuệ nhân tạo đã không còn là một cụm từ quá xa lạ đối với mọi người. Trí tuệ nhân tạo hay gọi tắt là AI đã len lỏi vào hầu hết các lĩnh vực như y tế, giáo dục, quân sự, nhà nước... AI là một bước đệm, là một sự thúc đẩy khiến con người đi nhanh và đi xa hơn nếu chúng ta biết sử dụng chúng đúng cách. AI cũng có nhiều loại như Natural Language Processing, Computer Vision, Robotics, ... Nhưng có lẽ quen thuộc nhất với mọi người đó chính là Data và các ứng dụng như phân tích và dự báo.

Trong nhiều lĩnh vực, việc phân tích dữ liệu là vô cùng cần thiết bởi nó đóng vai trò rất quan trọng cho sự phát triển sau này. Phân tích dữ liệu không chỉ là quá trình mô tả dữ liệu, mà còn là quá trình khám phá và tìm hiểu sâu hơn về các mẫu dữ liệu để đưa ra những quyết định thông minh và định hướng chiến lược. Việc áp dụng AI vào trong công việc và cuộc sống sẽ giúp con người ngày càng phát triển hơn.

Để có thể thực hiện việc phân tích dữ liệu, ta cần có những kiến thức cơ bản về việc phân tích. Tiếp đó là có được những mẫu dữ liệu cần phân tích để ta khai phá, tìm ra những giá trị và thông tin quý giá ẩn chứa trong chúng.

Sau khi thu thập dữ liệu, ta sẽ xác định dữ liệu đó phù hợp với các bài toán phân loại, hồi quy hay phân cụm, sau đó tiến hành các bước như tiền xử lý, chọn các mô hình phù hợp, đánh giá rồi đưa ra các dự báo dự theo một tập dữ liệu bất kỳ.

Phân tích dữ liệu có nhiều cách khác nhau để thực hiện, và có nhiều công cụ để hỗ trợ như Weka, Excel, PowerBI, ...nhưng không phải ai cũng có những kiến thức về những công cụ này nên bước đầu sẽ khó có thể thích ứng trong việc sử dụng. Lấy những kiến thức nền tảng từ việc phân tích dữ liệu bằng Python và áp dụng AI vào việc phát triển sản phẩm, ‘Phần mềm hỗ trợ phân tích dữ liệu và dự báo’ đã được tạo ra nhằm trợ giúp người dùng có thể tiếp cận

việc phân tích những bộ dữ liệu đơn giản bằng công cụ trên cho các bài toán phân loại và hồi quy. Phần mềm sẽ giúp người dùng có được những bước tiền đề trên con người theo đuổi đam mê phân tích và dự đoán dữ liệu.

Để đảm bảo kết quả của đề tài nghiên cứu, bài báo cáo được chia thành các phần như sau:

Chương 1: Tổng quan về phân tích dữ liệu

Trình bày tổng quan về AI, phân tích dữ liệu và bài toán dự báo, đồng thời nêu ra tình hình nghiên cứu bài toán dự báo trong nước và ở nước ngoài.

Chương 2: Các mô hình phân tích và dự báo

Trình bày các phương pháp kỹ thuật, cụ thể là phương pháp phân tích mô tả, dự báo, phương pháp phân tích hồi quy, phân loại và các công cụ có thể thực hiện bài toán.

Chương 3: Phân tích dữ liệu bằng công cụ Python

Trình bày phần thực nghiệm và đánh giá của dự án thông qua đầy đủ các bước từ tiền xử lý dữ liệu cho tới phân tích mô tả & bài toán dự báo. Từ đó đưa ra được các đánh giá và đề xuất phù hợp để cải thiện kết quả của dự án trong tương lai bằng ngôn ngữ Python.

Chương 4: Xây dựng phần mềm hỗ trợ phân tích dữ liệu

Trình bày quy trình xây dựng phần mềm, nêu rõ các chức năng có trong phần mềm và tiến hành áp dụng phần mềm trên một số bộ dữ liệu đã thu thập được.

Tổng kết lại, ta thấy đây là một đề tài khá rộng bởi nó có thể bao gồm nhiều kiến thức về AI, dữ liệu cũng như thiết kế hệ thống. Do vậy, ta cần thời gian dài để có thể nhận xét và đánh giá sản phẩm. Ngoài ra, ta còn cần phải tìm ra những khó khăn mà phần mềm chưa thực hiện được và đề xuất ra giải pháp phát triển trong tương lai.

CHƯƠNG 1 TỔNG QUAN VỀ PHÂN TÍCH DỮ LIỆU

1.1 Tổng quan về trí tuệ nhân tạo

1.1.1 Trí tuệ nhân tạo là gì

Trí tuệ nhân tạo hay trí thông minh nhân tạo (Artificial Intelligence – viết tắt là AI) là một ngành thuộc lĩnh vực khoa học máy tính (Computer science). Là trí tuệ do con người lập trình tạo nên với mục tiêu giúp máy tính có thể tự động hóa các hành vi thông minh như con người. Cụ thể, trí tuệ nhân tạo giúp máy tính có được những trí tuệ của con người như: biết suy nghĩ và lập luận để giải quyết vấn đề, biết giao tiếp do hiểu ngôn ngữ, tiếng nói, biết học và tự thích nghi, ... Công nghệ AI tạo ra máy móc và hệ thống thông minh thông qua việc sử dụng mô hình máy tính, kỹ thuật và công nghệ liên quan, giúp thực hiện các công việc yêu cầu trí thông minh của con người. Nhìn chung, đây là một ngành học rất rộng, bao gồm các yếu tố tâm lý học, khoa học máy tính và kỹ thuật. Một số ví dụ phổ biến về AI có thể kể đến ô tô tự lái, phần mềm dịch thuật tự động, trợ lý ảo trên điện thoại hay đối thủ ảo khi chơi trò chơi trên điện thoại.

1.1.2 Các loại trí tuệ nhân tạo

Công nghệ AI được chia làm 4 loại chính:

Loại 1: Công nghệ AI phản ứng

Công nghệ AI phản ứng có khả năng phân tích những động thái khả thi nhất của chính mình và của đối thủ, từ đó, đưa ra được giải pháp tối ưu nhất. Một ví dụ là Deep Blue, chương trình tự động chơi cờ vua của IBM đã đánh bại kì thủ thế giới Garry Kasparov vào những năm 1990. Công nghệ AI của Deep Blue có thể xác định các nước cờ và dự đoán những bước đi tiếp theo. Nó không có ký ức và không thể sử dụng những kinh nghiệm trong quá khứ để tiếp tục huấn luyện trong tương lai.

Loại 2: Công nghệ AI với bộ nhớ hạn chế

Các hệ thống AI này có thể sử dụng những kinh nghiệm trong quá khứ để đưa ra các quyết định trong tương lai. Một số chức năng ra quyết định này có mặt trong các loại thiết bị không người lái như xe, máy bay drone hoặc tàu ngầm. Kết hợp các cảm biến môi trường xung quanh công nghệ AI này có thể dự đoán được tình huống và đưa ra những bước hành động tối ưu cho thiết bị. Sau đó chúng sẽ được sử dụng để đưa ra hành động trong bước tiếp theo.

Loại 3: Công nghệ AI với lý thuyết trí tuệ nhân tạo

Công nghệ trí tuệ nhân tạo (AI) đang ngày càng tiến xa, và lý thuyết trí tuệ nhân tạo đóng vai trò quan trọng trong việc phát triển các ứng dụng và công nghệ mới. Lý thuyết trí tuệ nhân tạo là nền tảng cho việc hiểu và mô phỏng cách con người suy nghĩ, học và tương tác với môi trường xung quanh. Từ các phương pháp học máy và học sâu cho đến xử lý ngôn ngữ tự nhiên và thị giác máy, lý thuyết trí tuệ nhân tạo cung cấp khung nhìn và cơ sở cho việc phát triển các công nghệ AI tiên tiến. Bằng cách áp dụng các nguyên lý này, các nhà nghiên cứu và kỹ sư có thể xây dựng các hệ thống thông minh, có khả năng học và tự cải thiện một cách liên tục. Đồng thời, lý thuyết trí tuệ nhân tạo cũng giúp định hình các lĩnh vực như học tăng cường, nơi máy tính học từ trải nghiệm và phản hồi từ môi trường, mang lại tiềm năng đột phá trong việc tạo ra các hệ thống thông minh tự động và linh hoạt. Trong tương lai, sự kết hợp giữa công nghệ AI và lý thuyết trí tuệ nhân tạo hứa hẹn đem lại những tiến bộ to lớn, mở ra cánh cửa cho nhiều ứng dụng mới và đổi mới trong nhiều lĩnh vực khác nhau.

Loại 4: Công nghệ AI tự nhận thức

Công nghệ AI tự nhận thức, còn được gọi là trí tuệ nhân tạo tự nhận thức hoặc trí tuệ nhân tạo cường, là một lĩnh vực nghiên cứu và phát triển trong trí tuệ nhân tạo (AI). Điểm chính của công nghệ này là khả năng của máy tính hoặc hệ thống thông minh để tự hiểu, tự học và tự cải thiện mà không cần sự can thiệp từ con người. Trong các hệ thống AI truyền thống, các mô hình và thuật toán được lập trình để thực hiện các nhiệm vụ cụ thể dựa trên dữ liệu đầu

vào và quy tắc được xác định trước. Tuy nhiên, trong công nghệ AI tự nhận thức, máy tính có khả năng tự thích nghi và cải thiện khả năng của mình thông qua quá trình học máy không giám sát hoặc học tăng cường. Mục tiêu của AI tự nhận thức là tạo ra các hệ thống thông minh có khả năng nhận biết môi trường, tương tác với nó một cách tự nhiên, tự động hoạt động và thậm chí có khả năng tư duy giống như con người. Điều này đòi hỏi sự phát triển của các phương pháp học máy, xử lý ngôn ngữ tự nhiên, thị giác máy và các lĩnh vực khác của trí tuệ nhân tạo để tạo ra các hệ thống thông minh có khả năng học và tự cải thiện một cách liên tục.

1.1.3 Lợi ích và mặt trái của trí tuệ nhân tạo

Lợi ích của Trí thông minh nhân tạo

Có thể nói các trí tuệ nhân tạo AI không chỉ đơn thuần là một phần mềm máy tính có tính logic mà chúng còn chứa đựng cả trí tuệ của con người. Chúng biết suy nghĩ, lập luận để giải quyết các vấn đề, có thể giao tiếp với con người. Chính vì những tính năng vượt trội này mà AI có lợi ích vô cùng lớn.

- Phát hiện và ngăn chặn các rủi ro

AI giúp con người dự báo trước các rủi ro và mối nguy hại tiềm ẩn và hạn chế các thiệt hại đem lại. Các rủi ro được AI nhận biết như: Thảm họa thiên nhiên, động đất, sóng thần, núi lửa, dịch bệnh hay có mối nguy hại trong sản xuất kinh doanh.

- Hạn chế sử dụng sức lao động của con người

Nhờ quá trình học máy và tạo ra được các robot trong công nghiệp và đời sống. Con người sẽ không phải tốn nhiều sức lao động trong sản xuất, vận hành. Giờ đây, các máy móc robot sẽ thay con người làm việc đó.

- Xóa bỏ khoảng cách ngôn ngữ

Công nghệ AI sẽ giúp con người trên mọi Quốc gia có thể nói chuyện và hiểu nhau, thoả mái tiếp xúc. Có thêm nhiều cơ hội để học tập và làm việc trên khắp thế giới.

- Cá nhân hóa

Công nghệ AI sẽ đánh giá và thích ứng cũng như học hỏi đối tượng mà nó phục vụ. Từ đó, đưa ra phản ứng phù hợp nhất cho từng đối tượng riêng biệt.

Mặt trái của công nghệ trí tuệ nhân tạo

Trí tuệ nhân tạo (AI) đã mang lại nhiều lợi ích đáng kể cho xã hội, nhưng cũng không thiếu những mặt trái cần được nhìn nhận một cách thấu đáo. Một trong những mối lo ngại lớn nhất là về vấn đề việc làm. AI có khả năng thay thế con người trong nhiều lĩnh vực công việc, từ sản xuất công nghiệp đến dịch vụ tài chính, gây ra tình trạng thất nghiệp hàng loạt. Nhiều người lao động sẽ phải đổi mới với việc mất đi nguồn thu nhập ổn định và khó có thể tìm được việc làm mới trong bối cảnh công nghệ liên tục phát triển.

Một mặt trái khác của AI là vấn đề bảo mật và quyền riêng tư. Hệ thống AI ngày càng trở nên phức tạp và được sử dụng để thu thập, phân tích khối lượng lớn dữ liệu cá nhân. Điều này đặt ra nguy cơ về việc dữ liệu bị lạm dụng hoặc rơi vào tay kẻ xấu, dẫn đến những hậu quả nghiêm trọng cho người dùng. Hơn nữa, AI có thể bị lợi dụng để thực hiện các hành vi tội phạm như tấn công mạng, lừa đảo qua mạng và phát tán thông tin sai lệch.

Ngoài ra, AI cũng đặt ra những thách thức về đạo đức và trách nhiệm. Các hệ thống AI, dù thông minh đến đâu, vẫn hoạt động dựa trên các thuật toán và dữ liệu mà con người cung cấp. Điều này có nghĩa là AI có thể kế thừa và khuếch đại những thành kiến và định kiến có sẵn trong dữ liệu. Việc này dẫn đến các quyết định không công bằng hoặc phân biệt đối xử trong các lĩnh vực như tuyển dụng, xét duyệt tín dụng và thậm chí là trong hệ thống tư pháp.

Mặt khác, sự phát triển của AI cũng đẩy mạnh cuộc đua công nghệ giữa các quốc gia, đặc biệt là giữa các cường quốc kinh tế như Mỹ và Trung Quốc. Cuộc đua này không chỉ đòi hỏi nguồn lực tài chính khổng lồ mà còn dẫn đến nguy cơ về an ninh và hòa bình toàn cầu nếu AI được sử dụng cho mục đích quân sự.

Cuối cùng, sự phụ thuộc ngày càng nhiều vào AI có thể làm suy yếu khả năng tư duy và ra quyết định của con người. Khi các hệ thống AI đảm nhận nhiều nhiệm vụ quan trọng, con người có thể trở nên thụ động và mất đi kỹ năng tự giải quyết vấn đề. Điều này không chỉ ảnh hưởng đến cá nhân mà còn đến toàn xã hội, làm giảm khả năng sáng tạo và ứng phó với những thách thức mới. Tóm lại, mặc dù trí tuệ nhân tạo mang lại nhiều lợi ích vượt trội, chúng ta không thể bỏ qua những mặt trái tiềm ẩn của nó. Việc quản lý và phát triển AI cần phải được thực hiện một cách cẩn nhắc và có trách nhiệm, nhằm đảm bảo rằng công nghệ này phục vụ cho lợi ích của toàn nhân loại thay vì gây ra những hậu quả tiêu cực không mong muốn.

1.1.4 Các ứng dụng của trí tuệ nhân tạo

- Trong ngành y tế

Công nghệ AI làm thay đổi hoàn toàn bộ mặt ngành y tế. Có thể nói, y tế là lĩnh vực thiết thực nhất mà chúng ta quan tâm. Những ứng dụng của AI trong y học mang lại cho con người những giá trị đáng kinh ngạc. AI được sử dụng như một trợ lí chăm sóc sức khỏe cá nhân, chúng được sử dụng cho nghiên cứu và phân tích. Chúng có thể được sử dụng để lên lịch hẹn khám tại các cơ sở y tế, và điều quan trọng nhất chính là việc bệnh nhân được hỗ trợ 24/7. Bệnh nhân có thể dùng các app trên điện thoại chụp hình và điền các thông tin gửi lên một hệ thống trí tuệ nhân tạo và gần như tức thì kết quả chuẩn bệnh cũng như cách điều trị có thể được trả về. Hoặc một trong những ứng dụng nổi bật nhất của trí tuệ nhân tạo trong y tế chính là máy bay không người lái với tốc độ

nhanh hơn xe chuyên dụng lên đến 40%, thích hợp sử dụng cho những trường hợp cứu hộ khẩn cấp tại những vị trí có địa hình hiểm trở.

- Trong ngành giáo dục

Việc vận dụng trí tuệ nhân tạo trong các thao tác dạy và học, các trò chơi, phần mềm giáo dục giúp cải thiện và nâng cao trình độ học tập của con người. Ngoài ra, trí tuệ nhân tạo trong giáo dục còn có khả năng theo dõi sự tiến bộ của học sinh để giáo viên có thể biết và điều chỉnh cách dạy học sao cho hợp lý.

- Trong ngành vận tải

Trí tuệ nhân tạo được ứng dụng trong ngành vận tải thông qua những phương tiện giao thông vận tải tự lái, đặc biệt là ô tô đã đem lại những lợi ích kinh tế đáng kể nhờ khả năng cắt giảm chi phí và hạn chế những rủi ro tai nạn giao thông những vấn đề gây nguy hiểm đến tính mạng của con người.

- Trong ngành ngân hàng tài chính

Các ngân hàng, tổ chức tài chính đang sử dụng AI trong việc xử lý các hoạt động tài chính, tiền đầu tư và cổ phiếu, quản lý các tài sản khác nhau,... AI có thể vượt qua con người trong việc xử lý các giao dịch, giúp ngân hàng hỗ trợ khách hàng tốt hơn, cung cấp các giải pháp nhanh chóng hoặc nhận diện gương mặt của chủ tài khoản.

- Trong ngành dịch vụ

Công nghệ trí tuệ nhân tạo có khả năng nắm bắt được những thông tin về các hoạt động sử dụng dịch vụ của khách hàng thông qua việc thu thập và phân tích dữ liệu để từ đó đưa ra các giải pháp tối ưu, hiệu quả và phù hợp với nhu cầu sử dụng của họ. Điều này giúp ngành dịch vụ có thể hoạt động tốt hơn và mang lại những trải nghiệm thú vị, mới mẻ hơn cho người dùng. Chatbot chính là ví dụ điển hình cho ứng dụng này. Trong ngành truyền thông Đối với ngành

truyền thông, trí tuệ nhân tạo AI ra đời đã mang lại sự thay đổi lớn cho ngành trong việc tiếp cận các mục tiêu, đối tượng khách hàng tiềm năng. Dựa trên việc phân tích về nhân khẩu học, thói quen hoạt động trực tuyến hay những nội dung quảng cáo khách hàng hay xem để điều chỉnh thời gian và không gian cung cấp quảng cáo sao cho phù hợp.

- Trong ngành sản xuất

Nhà máy FANUC, Nhật bản là một trong những điển hình của việc ứng dụng AI vào trong sản xuất và sử dụng robot để sản xuất robot sản xuất ra 5000 robot mỗi tháng, sở hữu một trong những dây chuyền sản xuất vô cùng hiện đại của thế giới, dây chuyền tạo ra những thiết bị giúp chế tạo nhiều sản phẩm, từ ô tô cho đến điện thoại iPhone. Ở FANUC các robot tự xây dựng, giám sát và kiểm tra lẫn nhau. Trên thực tế, trí tuệ nhân tạo đã đưa chúng ta vào một chiều hướng mới vượt ra ngoài các bức tường của nhà máy FANUC. Robot đã chiếm lĩnh và tạo nên nhà máy thông minh một thời gian, robot ngày nay không còn thực hiện các nhiệm vụ cơ học, đơn điệu. Họ là những người tham gia thông minh trong Công nghiệp 4.0: cốt lõi là sự liên kết của nhà máy thực tế với thực tế ảo.

1.2 Tổng quan về phân tích dữ liệu

1.2.1 Phân tích dữ liệu là gì

Phân tích dữ liệu là một môn học tập trung vào việc rút ra những hiểu biết sâu sắc từ dữ liệu.

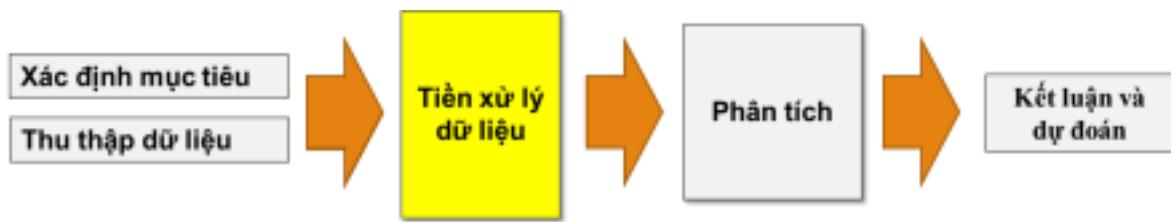
Phân tích dữ liệu là quá trình kiểm tra, làm sạch, chuyển đổi và mô hình hóa dữ liệu với mục tiêu khám phá thông tin hữu ích, đưa ra kết luận và hỗ trợ việc ra quyết định.

Phân tích dữ liệu được áp dụng rộng rãi trong nhiều lĩnh vực và ngành nghề, bao gồm khoa học dữ liệu, kinh doanh, tài chính, y tế, marketing, hành chính công, và nhiều lĩnh vực khác. Qua việc phân tích dữ liệu, người ta có thể

tìm ra xu hướng, mẫu số, tương quan và những thông tin quan trọng khác từ dữ liệu, từ đó hỗ trợ quyết định và lập kế hoạch trong các hoạt động kinh doanh và nghiên cứu.

Các phương pháp và công cụ phân tích dữ liệu có thể bao gồm việc sử dụng các thuật toán thống kê, khai phá dữ liệu, học máy (machine learning), trực quan hóa dữ liệu, mô hình hóa, và các phương pháp khác để tạo ra thông tin có ý nghĩa từ dữ liệu thu thập được.

1.2.2 Quy trình phân tích dữ liệu



Hình 1.1: Quy trình phân tích dữ liệu

Quy trình phân tích dữ liệu thường bao gồm các bước chính:

Xác định mục tiêu và thu thập dữ liệu:

- Xác định mục tiêu:** là những kết quả cụ thể mà ta muốn đạt được thông qua việc xử lý và phân tích dữ liệu. Mục tiêu này xác định hướng đi và phạm vi của quá trình phân tích, giúp ta tập trung vào việc thu thập thông tin quan trọng và thực hiện các phân để đáp ứng các yêu cầu hoặc nhu cầu cụ thể.
- Thu thập dữ liệu:** là thu thập dữ liệu từ các nguồn khác nhau như cơ sở dữ liệu, tệp tin, trang web, thiết bị cảm biến, và nhiều nguồn khác. Dữ liệu có thể là số liệu, văn bản, hình ảnh, hoặc âm thanh.

Tiền xử lý dữ liệu: Dữ liệu thường không hoàn hảo và có thể chứa nhiều, dữ liệu bị thiếu, hoặc không chính xác. Tiền xử lý dữ liệu bao gồm việc tóm lược dữ liệu, làm sạch dữ liệu, tích hợp dữ liệu, chuyển đổi dữ liệu, rút gọn dữ liệu và rời rạc hóa dữ liệu để chuẩn bị cho bước phân tích.

Phân tích dữ liệu: Bước quan trọng này dựa vào kiến thức và kỹ thuật phân tích để tìm ra mối liên hệ và thông tin ẩn sau dữ liệu. Phân tích dữ liệu có thể sử dụng các phương pháp phân tích mô tả, phân tích hồi quy, phân tích sự khác biệt, thống kê, machine learning, data mining, và nhiều kỹ thuật khác.

Kết luận và dự đoán: Dựa trên phân tích và thông tin từ dữ liệu, chúng ta có thể rút ra kết luận, hiểu rõ hơn về tình hình, và thậm chí đưa ra dự đoán cho tương lai.

1.3 Tổng quan về bài toán dự báo

1.3.1 Lịch sử bài toán dự báo

Bài toán dự báo có một lịch sử lâu đời và đã phát triển qua nhiều giai đoạn. Dưới đây là một cái nhìn tổng quan về lịch sử hình thành của bài toán dự báo:

- Thời kỳ tiền Công nghiệp (Trước thế kỷ 18): Trong giai đoạn này, con người thường dự báo dựa trên kinh nghiệm và tri thức truyền đạt qua thế hệ. Dự báo chủ yếu dựa trên sự quan sát của thiên văn học, thời tiết, và các hiện tượng tự nhiên.

- Cách mạng Công nghiệp và thống kê (Thế kỷ 18 - 19): Trong thời kỳ này, việc sử dụng số liệu và thống kê để dự báo đã trở nên phổ biến hơn. Những ý tưởng về xác suất và phân phối bắt đầu được áp dụng vào việc dự báo.

- Thế kỷ 20 và Kỹ thuật số hóa: Sự phát triển của máy tính và kỹ thuật số hóa đã mở ra những cơ hội mới trong việc dự báo. Các phương pháp thống kê, mô hình hóa toán học, và kỹ thuật machine learning bắt đầu được sử dụng rộng rãi để dự báo trong nhiều lĩnh vực.

- Thống kê Bayes và Kỹ thuật Machine learning (Thế kỷ 20 - 21): Thống kê Bayes và các kỹ thuật machine learning như học máy, học sâu, và học tăng cường đã thúc đẩy khả năng dự báo thông qua việc xử lý dữ liệu phức tạp và tìm ra các mẫu ẩn.

- Dự báo trong thời đại số hóa (Hiện nay): Với sự gia tăng mạnh mẽ về khả năng tính toán, khối lượng dữ liệu khổng lồ, và sự phát triển của trí tuệ nhân tạo, bài toán dự báo đang trở nên càng quan trọng và phức tạp hơn. Các công nghệ mới như big data analytics, deep learning, và dự báo dựa trên mạng xã hội đang mở ra nhiều cơ hội và thách thức mới trong lĩnh vực này.

- Trong suốt quá trình phát triển, bài toán dự báo đã chuyển từ việc dự đoán dựa trên sự quan sát đơn thuần đến việc sử dụng các phương pháp phức tạp để xác định mối quan hệ phức hợp và xu hướng từ dữ liệu. Lịch sử hình thành này thể hiện sự tiến bộ và tầm quan trọng của bài toán dự báo trong việc hỗ trợ quyết định và phát triển trong nhiều lĩnh vực.

Bài toán dự báo là một trong những thách thức quan trọng trong lĩnh vực phân tích dữ liệu, nơi chúng ta cố gắng dự đoán giá trị của một biến mục tiêu trong tương lai dựa trên dữ liệu lịch sử và các yếu tố ảnh hưởng. Mục tiêu chính của bài toán dự báo là xây dựng một mô hình có khả năng hiểu và ứng dụng các mẫu, xu hướng và quy luật từ dữ liệu để thực hiện việc dự đoán một cách chính xác và đáng tin cậy.

Trong thời đại số hóa hiện nay, bài toán dự báo đối mặt với những thách thức và cơ hội mới. Sự phát triển của big data analytics cho phép thu thập và xử lý khối lượng lớn dữ liệu từ nhiều nguồn khác nhau, tạo ra cơ hội để tìm ra các mẫu và thông tin quan trọng từ dữ liệu này. Dự báo dựa trên mạng xã hội là một lĩnh vực mới nổi trong đó dữ liệu từ các nền tảng mạng xã hội được sử dụng để dự báo hành vi, xu hướng và tương tác của người dùng.

Tổng quan về lịch sử hình thành của bài toán dự báo cho thấy sự tiến bộ và quan trọng của nó trong việc hỗ trợ quyết định và phát triển trong nhiều lĩnh vực. Bài toán dự báo đã phát triển từ việc dự đoán dựa trên kinh nghiệm và quan sát đơn thuần đến việc sử dụng các phương pháp và công nghệ phức tạp để tìm hiểu và dự đoán dựa trên dữ liệu lịch sử.

1.3.2 Tình hình phát triển của bài toán dự báo ở Việt Nam

Bài toán dự báo có sự ảnh hưởng to lớn tại cả Việt Nam. Dự báo giúp cải thiện quản lý, định hình chiến lược, và tối ưu hóa tài nguyên trong nhiều lĩnh vực. Có một số điểm đáng chú ý về tình hình phân tích dữ liệu tại Việt Nam:

- Phát triển đang ở giai đoạn đầu: Trong một số lĩnh vực, bài toán dự báo tại Việt Nam đang ở giai đoạn đầu của sự phát triển. Việc áp dụng các phương pháp phân tích dữ liệu và dự báo mới còn đang được tìm hiểu và thí nghiệm.

- Ứng dụng trong nông nghiệp và kinh tế: Tại Việt Nam, dự báo có ứng dụng quan trọng trong nông nghiệp, nhằm dự đoán thời tiết, mùa màng, và nhu cầu năng lượng. Nó cũng được áp dụng trong kinh tế, dự báo tăng trưởng GDP, lạm phát, và tỷ giá.

- Thách thức từ dữ liệu: Một thách thức cho việc dự báo tại Việt Nam là khả năng thu thập và quản lý dữ liệu chất lượng. Dữ liệu thường không đầy đủ và có thể gặp vấn đề về tính nhất quán và độ tin cậy.

Dưới đây là một cái nhìn tổng quan về phát triển của bài toán dự báo ở Việt Nam:

1. Thời kỳ tiền Công nghiệp và Cách mạng Công nghiệp: Trước thế kỷ 18 và trong giai đoạn Cách mạng Công nghiệp, dự báo ở Việt Nam cũng dựa trên các quan sát và tri thức truyền đạt qua thế hệ, tương tự như các nước khác. Những thông tin về thời tiết, thiên văn học và các hiện tượng tự nhiên được sử dụng để dự báo.

2. Thế kỷ 20 và Kỹ thuật số hoá: Với sự phát triển của máy tính và kỹ thuật số hoá, Việt Nam đã bắt kịp xu hướng sử dụng các phương pháp thống kê và mô hình hóa toán học để dự báo. Các ngành công nghiệp như tài chính, thương mại và sản xuất đã áp dụng các phương pháp này để dự báo xu hướng thị trường, tiêu thụ và sản xuất.

3. Thông kê Bayes và Kỹ thuật Machine learning: Thông kê Bayes và các kỹ thuật machine learning như học máy và học sâu đã được áp dụng rộng rãi trong bài toán dự báo ở Việt Nam. Các công ty và tổ chức nghiên cứu đã sử dụng các phương pháp này để dự báo trong lĩnh vực tài chính, thương mại, y tế và nông nghiệp.

4. Sự phát triển của big data và dự báo dựa trên mạng xã hội: Việt Nam cũng đã nhận thấy tiềm năng của big data và dữ liệu từ mạng xã hội trong bài toán dự báo. Việc thu thập và phân tích dữ liệu từ các nguồn khác nhau như mạng xã hội, thiết bị cảm biến và các hệ thống thông tin công nghệ cao đang mở ra nhiều cơ hội mới trong việc dự báo tình hình và xu hướng.

1.3.3 Tình hình phát triển của bài toán dự báo ở thế giới

Trong thế giới ngày nay, bài toán dự báo đang trở nên ngày càng quan trọng và phô biến trong nhiều lĩnh vực khác nhau, từ kinh doanh đến y tế và khoa học. Dưới đây là một số xu hướng và tình hình phát triển của bài toán dự báo ở thế giới:

- Sử dụng trí tuệ nhân tạo và học máy.

Trí tuệ nhân tạo (AI) và học máy đang đóng vai trò quan trọng trong việc phát triển các mô hình dự báo. Công nghệ này cho phép chúng ta xử lý các tập dữ liệu lớn và phức tạp, đồng thời cải thiện hiệu suất dự báo.

- Dự báo thời tiết.

Trong lĩnh vực thời tiết, việc dự báo chính xác và đáng tin cậy ngày càng trở nên quan trọng hơn do ảnh hưởng lớn của thời tiết đến nhiều hoạt động, từ nông nghiệp đến giao thông và du lịch. Công nghệ mới như hệ thống dự báo thời tiết tự động và mô hình học sâu đang được phát triển để cải thiện chính xác của các dự báo.

- Dự báo tài chính.

Trong lĩnh vực tài chính, việc dự báo xu hướng thị trường, giá cổ phiếu và các chỉ số tài chính khác là rất quan trọng để đưa ra các quyết định đầu tư thông minh. Các mô hình học máy và phân tích dữ liệu đang được áp dụng rộng rãi để dự báo và quản lý rủi ro trong các thị trường tài chính.

- Dự báo y tế.

Trong lĩnh vực y tế, việc dự báo các dịch bệnh, xu hướng sức khỏe cộng đồng và kết quả điều trị cá nhân có thể cứu sống hàng triệu người. Các công nghệ mới như phân tích dữ liệu di truyền và học máy đang được sử dụng để xây dựng các mô hình dự báo y tế tiên tiến.

- Dự báo về môi trường.

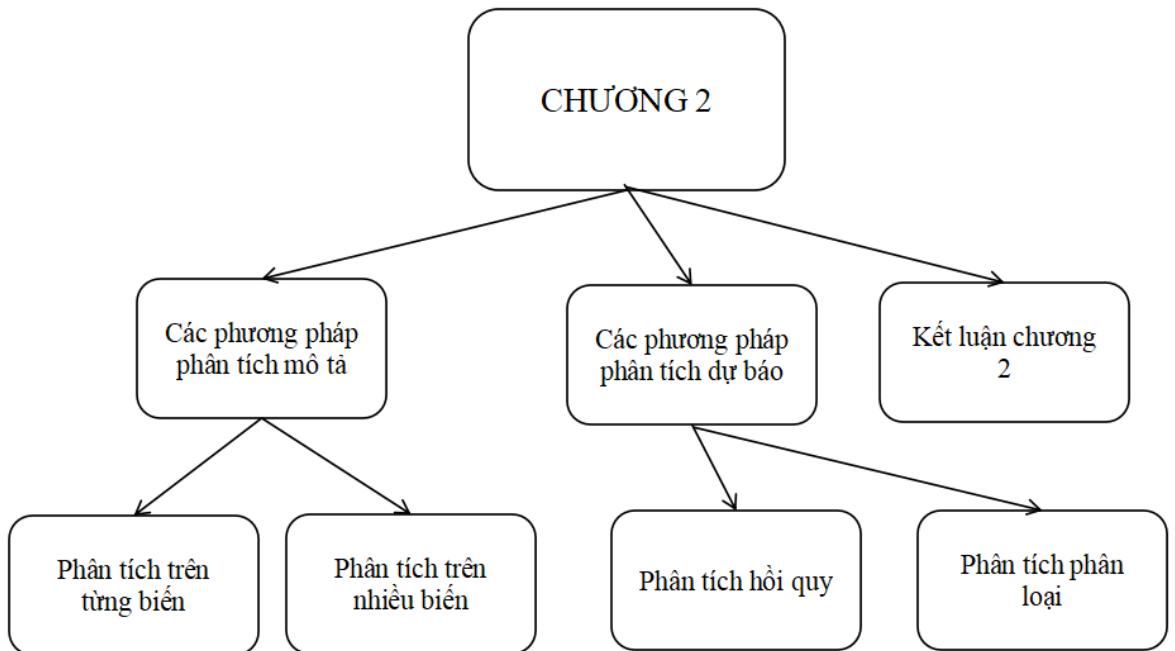
Trong mối quan tâm ngày càng tăng về biến đổi khí hậu và bảo vệ môi trường, việc dự báo và đánh giá các tác động của các biến đổi môi trường là rất quan trọng. Công nghệ dự báo môi trường đang phát triển để cung cấp thông tin cần thiết cho các nhà quản lý môi trường và chính phủ.

Tổng quan lại, bài toán dự báo đang trở thành một lĩnh vực nghiên cứu và ứng dụng rộng lớn, với nhiều ứng dụng quan trọng trong nhiều lĩnh vực khác nhau của cuộc sống hàng ngày. Sự phát triển của công nghệ và khả năng xử lý dữ liệu lớn đang mở ra nhiều cơ hội mới cho việc cải thiện chính xác và hiệu suất của các mô hình dự báo.

CHƯƠNG 2 CÁC PHƯƠNG PHÁP PHÂN TÍCH MÔ TẢ VÀ DỰ BÁO

2.1 Giới thiệu tổng quan chương 2

Trong chương 2, chúng ta sẽ cùng tìm hiểu về các phương pháp phân tích mô tả và phân tích dự báo. Trong phương pháp phân tích mô tả, ta tìm hiểu về phân tích trên từng biến mà nhiều biến. Về phương pháp phân tích dự báo, ta tìm hiểu về các phương pháp phân tích hồi quy và phân tích dự báo, các mô hình và thuật toán có trong mỗi loại phân tích. Cuối cùng là kết luận chung cho chương 2.



Hình 2.1: Sơ đồ tổng quát chương 2

2.2 Phương pháp phân tích mô tả

2.2.1 Phân tích mô tả

Phân tích mô tả là một phương pháp trong lĩnh vực thống kê và phân tích dữ liệu, nhằm mô tả và tóm tắt các đặc điểm chính của một tập dữ liệu một cách dễ hiểu và ngắn gọn. Mục tiêu của phân tích mô tả là giúp hiểu sâu hơn về dữ liệu mà chúng ta đang làm việc, nhận ra các đặc trưng quan trọng, và cung cấp một cái nhìn tổng quan về phân phối và biến đổi của dữ liệu.

Phân tích mô tả thường bao gồm các khía cạnh sau:

- **Thông kê tóm tắt:** Đây là các số liệu thống kê cơ bản như trung bình, trung vị, độ lệch chuẩn, và phân vị. Các số liệu này giúp ta hiểu về trung tâm và phân tán của dữ liệu.
- **Biểu đồ:** Biểu đồ thường được sử dụng để biểu diễn dữ liệu một cách trực quan. Các biểu đồ như biểu đồ cột, biểu đồ đường, biểu đồ hình tròn, và biểu đồ hộp giúp ta thấy được sự phân bố và xu hướng của dữ liệu.
- **Phân phối dữ liệu:** Phân tích phân phối dữ liệu giúp ta hiểu về tỷ lệ xuất hiện của các giá trị khác nhau trong tập dữ liệu. Điều này có thể làm bằng cách tạo biểu đồ phân phối tần số hoặc xây dựng biểu đồ kernel density.
- **Kiểm tra sự tương quan:** Phân tích mô tả cũng có thể liên quan đến việc kiểm tra sự tương quan giữa các biến. Điều này có thể thực hiện bằng cách sử dụng biểu đồ tương quan hoặc tính toán hệ số tương quan Pearson.
- **Xác định điểm ngoại lệ:** Phân tích mô tả cũng giúp xác định các điểm dữ liệu ngoại lệ, tức là những giá trị rất khác biệt so với phần còn lại của dữ liệu.
- **Tổng kết và nhận xét:** Cuối cùng, phân tích mô tả thường đi kèm với việc tổng kết và nhận xét về các đặc điểm quan trọng của dữ liệu, những mẫu thú vị, và những điểm mạnh và điểm yếu của tập dữ liệu.

Phân tích mô tả giúp xây dựng một cái nhìn sâu hơn về tập dữ liệu ban đầu và tạo nền tảng cho các phân tích tiếp theo như dự báo, phân tích hồi quy, hay machine learning.

2.2.2 Phương pháp phân tích trên từng biến

Khi thực hiện phân tích trên một biến (hoặc một thuộc tính), mục tiêu chính là hiểu rõ các đặc điểm cơ bản của biến đó. Điều này thường bao gồm xác định và xử lý các giá trị ngoại lai hoặc bất thường (Outliers). Đây là các giá trị dữ liệu mà rất khác biệt so với phần lớn các giá trị khác trong tập dữ liệu. Các giá trị ngoại lai có thể xuất hiện do lỗi nhập liệu, lỗi đo lường, hoặc đơn giản là do các sự kiện hiếm gặp.

Việc xác định các Outliers có vai trò quan trọng và là mắt xích liên kết giữa phân tích mô tả và phân tích hồi quy, bởi vì ta có thể tiến hành làm sạch những giá trị này tại công đoạn tiền xử lý dữ liệu của phân tích hồi quy. Cụ thể với từng loại dữ liệu khác nhau, ta sẽ phân tích như sau:

Dữ liệu số:

- **Biểu đồ Histogram:** Biểu đồ hiển thị tần suất xuất hiện của các khoảng giá trị dữ liệu.
- **Các đại lượng thống kê:** Bao gồm mean (trung bình), stdev (độ lệch chuẩn), median (trung vị), quartile (phân vị) ... Các giá trị này giúp mô tả trung bình, phương sai và phân phối của dữ liệu.
- **Biểu đồ Box & Whisker (Boxplot):** Biểu đồ hiển thị tổng quan giá trị đó bao gồm các giá trị đại lượng thống kê đã tính được.

Dữ liệu phi số:

- **Bảng tần suất (Frequency table):** Biểu đồ liệt kê các giá trị khác nhau của biến và số lần xuất hiện của mỗi giá trị.

- **Biểu đồ cột (Bar chart):** Biểu đồ thể hiện tần suất của từng giá trị dữ liệu dưới dạng các cột đứng.

- **Biểu đồ hình tròn hoặc donut (Pie chart, Donut chart):** Biểu đồ thể hiện phần trăm tần suất của từng giá trị trong tổng số.

2.2.3 Phương pháp phân tích trên nhiều biến

Phân tích trên nhiều biến hướng tới việc hiểu mối quan hệ và tương tác giữa các biến trong tập dữ liệu. Điều này có thể giúp bạn phát hiện ra các mẫu, xu hướng hoặc tương quan có thể tồn tại giữa chúng.

Các mối liên hệ giữa các biến (Interrelationships) có thể là nhiều dạng khác nhau: Mối tương quan tuyến tính, tương quan không tuyến tính, tương quan ngược... Với mỗi mối liên hệ, ta có thể phân tích và tìm ra được cách các biến tương tác và ảnh hưởng lẫn nhau.

Việc phân tích trên nhiều biến cũng có mối liên hệ mật thiết đến phân tích hồi quy khi giúp ta xác định được các giá trị ngoại lai của dữ liệu. Do là phân tích nhiều biến, vậy nên sẽ có 3 kiểu dữ liệu phân tích khác nhau: Số, phi số và hỗn hợp (cả số và phi số):

Dữ liệu số:

+ **Scatter Plot (Biểu đồ Scatter):** Biểu đồ thể hiện mối quan hệ giữa hai biến số. Mỗi điểm trên biểu đồ thể hiện một cặp giá trị của hai biến trên trực ngang và dọc. Biểu đồ này dùng để tìm kiếm sự tương quan giữa 2 biến số như tương quan tuyến tính hoặc không tuyến tính.

+ **Bảng dữ liệu thống kê (Statistical Summary Table):** Tạo bảng để liệt kê các đại lượng thống kê (mean, median, stdev...) giữa các biến số của dữ liệu.

Dữ liệu phi số:

+ **Bảng dữ liệu thống kê (Statistical Summary Table):** Cũng là bảng dữ liệu thống kê nhưng với giá trị phi số, đó sẽ chỉ có giá trị tần suất xuất hiện (mode) của dữ liệu.

Dữ liệu hỗn hợp:

+ **Bảng thống kê tổng hợp:** Đây là sự kết hợp giữa bảng dữ liệu thống kê của dữ liệu số và phi số. Sự kết hợp tổng quan này sẽ cho ta bao quát được phân bổ của dữ liệu.

+ **Biểu đồ Box-and-Whisker (Boxplot):** Được sử dụng để so sánh phân phối của một dữ liệu số với tần suất của một dữ liệu phi số. Biểu đồ này sẽ cho ta mối quan hệ mật thiết về sự ảnh hưởng của các giá trị phi số lên giá trị số được phân tích.

2.3 Phương pháp phân tích dự báo

2.3.1 Tổng quan về phân tích hồi quy

Phân tích hồi quy là một tập hợp các phương pháp thống kê được sử dụng để ước tính các mối quan hệ giữa một biến phụ thuộc và một hoặc nhiều biến độc lập. Nó có thể được sử dụng để đánh giá sức mạnh của mối quan hệ giữa các biến và để mô hình hóa mối quan hệ trong tương lai giữa chúng.

Phân tích hồi quy là một cách phân loại toán học để xác định biến nào trong số những biến đó thực sự có tác động. Nó trả lời các câu hỏi: Yếu tố nào quan trọng nhất? Cái nào có thể bỏ qua? Các yếu tố đó tương tác với nhau như thế nào? Và quan trọng nhất, chúng ta chắc chắn như thế nào về tất cả những yếu tố này?

Trong phân tích hồi quy, ta cần xác định một biến phụ thuộc – yếu tố chính mà ta đang cố gắng hiểu hoặc dự đoán. Phân tích hồi quy bao gồm một số biến thể, chẳng hạn như tuyến tính, nhiều tuyến tính và phi tuyến tính. Trong đó mô hình phổ biến là tuyến tính và nhiều tuyến tính. Đối với phân tích hồi quy phi

tuyến, chúng thường được sử dụng cho các tập dữ liệu phức tạp hơn trong đó các biến phụ thuộc và độc lập thể hiện mối quan hệ phi tuyến.

2.3.2 Các phương pháp phân tích hồi quy

Để phân tích hồi quy có rất nhiều phương pháp để phân tích. Dưới đây sẽ là một số phương pháp quan trọng dùng để phân tích hồi quy:

- Hồi quy tuyến tính (Linear Regression): Hồi quy tuyến tính giả định rằng có một mối quan hệ tuyến tính giữa biến phụ thuộc (Y) và các biến độc lập (X). Ý tưởng chính là sử dụng các giá trị của các biến độc lập để dự đoán giá trị của biến phụ thuộc. Mục tiêu của hồi quy tuyến tính là tìm ra các hệ số hồi quy sao cho hàm dự đoán tuyến tính đạt được sai số nhỏ nhất.

Công thức:

Hàm dự đoán tuyến tính có thể được biểu diễn bằng công thức sau:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \varepsilon$$

Trong đó:

- Y là biến phụ thuộc (giá trị dự đoán)

- X_1, X_2, \dots, X_n là các biến độc lập

- $\beta_0, \beta_1, \beta_2, \dots, \beta_n$ là các hệ số hồi quy (slope)

- ε là sai số ngẫu nhiên

Các loại hồi quy tuyến tính:

Hồi quy tuyến tính đơn biến: Chỉ có một biến độc lập (X) và một biến phụ thuộc (Y). Đây là trường hợp đơn giản nhất của hồi quy tuyến tính. **Hồi quy tuyến tính đa biến:** Có nhiều hơn một biến độc lập (X_1, X_2, \dots, X_n) và một biến phụ thuộc (Y). Trong trường hợp này, mô hình hồi quy sẽ có nhiều hơn một hệ số hồi quy. **Phương pháp ước lượng hệ số:** Có nhiều phương pháp để ước lượng các hệ số hồi quy, bao gồm:

Phương pháp bình phương tối thiểu (OLS): Đây là phương pháp thông dụng nhất để ước lượng các hệ số hồi quy. Nó tìm ra các giá trị của các hệ số sao cho tổng bình phương của sai số là nhỏ nhất.

Phương pháp Gradient descent: Đây là một phương pháp tối ưu để tìm ra các giá trị của các hệ số bằng cách điều chỉnh chúng dựa trên đạo hàm của hàm mất mát.

Kiểm định và đánh giá:

Sau khi ước lượng các hệ số, cần kiểm tra mô hình để xác định tính chính xác và ý nghĩa của nó. Các phương pháp kiểm định và đánh giá bao gồm kiểm tra giả thuyết, kiểm tra F, kiểm tra t, R-square và RMSE (Root Mean Squared Error).

Các yếu điểm:

Mặc dù hồi quy tuyến tính là một công cụ mạnh trong phân tích thống kê, nó cũng có một số yếu điểm:

Giả định về mối quan hệ tuyến tính: Hồi quy tuyến tính giả định rằng mỗi quan hệ giữa biến phụ thuộc và biến độc lập là tuyến tính. Nếu mối quan hệ không tuân theo mô hình tuyến tính, kết quả có thể không chính xác.

Quan sát và xử lý nhiễu: Nhiều trong dữ liệu có thể ảnh hưởng đến kết quả của mô hình. Cần kiểm tra và xử lý nhiễu để đảm bảo tính chính xác của kết quả.

- Hồi quy Ridge (Ridge Regression): Hồi quy Ridge là phiên bản cải tiến của hồi quy tuyến tính bằng cách thêm hệ số điều chỉnh L_2 vào hàm mất mát. Điều này giúp kiểm soát độ phức tạp của mô hình và tránh tình trạng quá khớp (overfitting). Tuy ưu điểm là giảm overfitting và xử lý đa cộng tuyến, nhưng cần lựa chọn tham số điều chỉnh chính xác.

- Hồi quy Lasso (Lasso Regression): Hồi quy Lasso cũng cải tiến từ hồi quy tuyến tính, nhưng thay vì l2, nó sử dụng hệ số điều chuẩn l1 để thúc đẩy một số hệ số về 0. Điều này dẫn đến lựa chọn biến tự động và giảm biến quan trọng. Lasso giải quyết vấn đề "chọn biến" nhưng cần phải có tham số điều chuẩn chính xác.

2.3.3 Tổng quan về phân tích phân loại

Phân tích phân loại là một phương pháp quan trọng trong lĩnh vực khoa học dữ liệu và máy học, giúp tổ chức và phân loại các đối tượng vào các nhóm khác nhau dựa trên các đặc điểm hoặc thuộc tính của chúng. Các khái niệm cơ bản như thuộc tính, nhãn, và mô hình phân loại là các yếu tố quan trọng trong quá trình này. Bằng cách sử dụng các thuật toán và kỹ thuật phân loại, chúng ta có thể xây dựng các mô hình có khả năng dự đoán nhãn cho các mẫu mới với độ chính xác cao. Qua quá trình huấn luyện và kiểm định, chúng ta có thể đánh giá và cải thiện hiệu suất của mô hình, từ đó áp dụng phân tích phân loại vào nhiều lĩnh vực thực tiễn như nhận dạng hình ảnh, dự đoán tín dụng, hoặc phân loại văn bản.

2.3.4 Các phương pháp phân tích phân loại

Để phân tích phân loại có rất nhiều phương pháp để phân tích. Dưới đây sẽ là một số phương pháp quan trọng dùng để phân tích phân loại.

- Hồi quy Logistic (Logistic Regression): Hồi quy logistic là một phương pháp phân tích thống kê được sử dụng rộng rãi để dự đoán xác suất của một sự kiện nhị phân dựa trên các biến độc lập. Dù tên gọi của nó có chứa từ "hồi quy", hồi quy logistic thực ra là một thuật toán phân loại, không phải là một phương pháp hồi quy truyền thống. Phương pháp này thường được áp dụng khi biến mục tiêu hoặc nhãn chỉ nhận một trong hai giá trị, thường là 0 và 1. Hồi quy logistic sử dụng hàm logistic để biểu diễn xác suất của sự kiện xảy ra, đảm bảo rằng giá trị xác suất luôn nằm trong khoảng từ 0 đến 1, phù hợp với tính chất xác suất. Thuật toán tối ưu hóa các tham số của mô hình bằng cách sử dụng

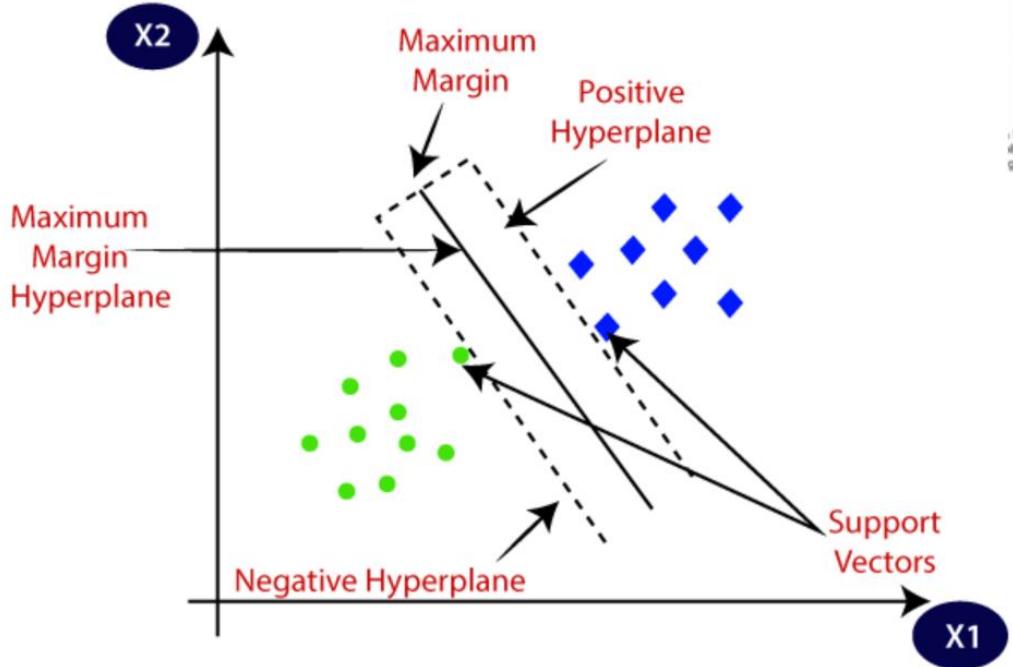
phương pháp cực đại ước lượng hợp lý, mục tiêu là tìm ra bộ tham số tối ưu để mô hình có khả năng dự đoán xác suất tốt nhất. Hồi quy logistic là một công cụ mạnh mẽ và linh hoạt được sử dụng rộng rãi trong nhiều lĩnh vực như y học, tài chính, marketing và khoa học xã hội.

- SVM (Support Vector Machine):

Hỗ trợ Vector Machine hay SVM là một trong những thuật toán Học có Giám sát phổ biến nhất, được sử dụng cho các bài toán Phân loại cũng như Hồi quy. Tuy nhiên, chủ yếu, nó được sử dụng cho các vấn đề Phân loại trong Học máy.

Mục tiêu của thuật toán SVM là tạo đường hoặc ranh giới quyết định tốt nhất có thể tách không gian n chiều thành các lớp để chúng ta có thể dễ dàng đặt điểm dữ liệu mới vào đúng danh mục trong tương lai. Ranh giới quyết định tốt nhất này được gọi là siêu phẳng.

SVM chọn các điểm / vec-tor cực hạn giúp tạo siêu phẳng. Những trường hợp cực đoan này được gọi là vec-tor hỗ trợ, và do đó thuật toán được gọi là Máy vector hỗ trợ. Hãy xem xét sơ đồ dưới đây, trong đó có hai danh mục khác nhau được phân loại bằng cách sử dụng ranh giới quyết định hoặc siêu phẳng:



Hình 2.2: Thuật toán máy hỗ trợ vec-tor (SVM).

- Giải thích về SVM

Máy vec-tơ hỗ trợ là một thuật toán học có giám sát để sắp xếp dữ liệu thành hai loại. Nó được đào tạo với một loạt dữ liệu đã được phân loại thành hai loại, xây dựng mô hình như nó được đào tạo ban đầu. Nhiệm vụ của thuật toán SVM là xác định loại điểm dữ liệu mới thuộc về loại nào. Điều này làm cho SVM trở thành một loại bộ phân loại tuyến tính không nhị phân.

Một thuật toán SVM không chỉ nên đặt các đối tượng vào các danh mục mà còn phải đặt lè giữa chúng trên một biểu đồ càng rộng càng tốt.

Một số ứng dụng của SVM bao gồm:

- Phân loại văn bản và siêu văn bản.
- Phân loại hình ảnh.
- Nhận dạng các ký tự viết tay.
- Khoa học sinh học, bao gồm phân loại protein.

- Các loại thuật toán hỗ trợ máy vec-tơ
- **SVM tuyến tính:** SVM tuyến tính được sử dụng cho dữ liệu có thể phân tách tuyến tính, có nghĩa là nếu một tập dữ liệu có thể được phân loại thành hai lớp bằng cách sử dụng một đường thẳng duy nhất, thì dữ liệu đó được gọi là dữ liệu có thể phân tách tuyến tính và bộ phân loại được sử dụng gọi là bộ phân loại SVM tuyến tính.
- **SVM phi tuyến tính:** SVM phi tuyến tính được sử dụng cho dữ liệu được phân tách không theo tuyến tính, có nghĩa là nếu tập dữ liệu không thể được phân loại bằng cách sử dụng một đường thẳng, thì dữ liệu đó được gọi là dữ liệu phi tuyến tính và bộ phân loại được sử dụng được gọi là Không bộ phân loại SVM tuyến tính.

● Siêu phẳng và Vectơ hỗ trợ trong thuật toán SVM

Siêu phẳng: Có thể có nhiều đường / ranh giới quyết định để phân tách các lớp trong không gian n chiều, nhưng chúng ta cần tìm ra ranh giới quyết định tốt nhất giúp phân loại các điểm dữ liệu. Ranh giới tốt nhất này được gọi là siêu phẳng của SVM.

Kích thước của siêu phẳng phụ thuộc vào các tính năng có trong tập dữ liệu, có nghĩa là nếu có 2 đặc điểm (như trong hình) thì siêu phẳng sẽ là một đường thẳng. Và nếu có 3 đặc điểm thì siêu phẳng sẽ là mặt phẳng 2 chiều.

Chúng tôi luôn tạo siêu phẳng có lề tối đa, nghĩa là khoảng cách tối đa giữa các điểm dữ liệu.

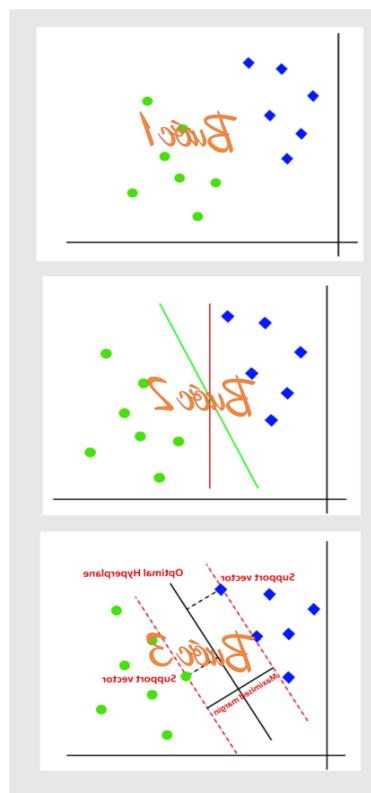
Hỗ trợ Vectơ:

Các điểm dữ liệu hoặc vectơ gần nhất với siêu phẳng và ảnh hưởng đến vị trí của siêu phẳng được gọi là Vectơ hỗ trợ. Vì những vectơ này hỗ trợ siêu phẳng, do đó được gọi là vectơ Hỗ trợ.

● SVM hoạt động như thế nào?

SVM tuyến tính:

Hoạt động của thuật toán SVM có thể được hiểu bằng cách sử dụng một ví dụ. Giả sử chúng ta có một tập dữ liệu có hai thẻ (xanh lá cây và xanh lam), và tập dữ liệu có hai đặc điểm x_1 và x_2 . Chúng tôi muốn một bộ phân loại có thể phân loại cặp tọa độ (x_1, x_2) theo màu xanh lục hoặc xanh lam. Hãy xem xét hình ảnh dưới đây:



Hình 2.3: Thuật toán SVM tuyến tính.

Vì nó là không gian 2 chiều nên chỉ cần sử dụng một đoạn thẳng, chúng ta có thể dễ dàng tách hai lớp này. Nhưng có thể có nhiều dòng có thể phân tách các lớp này. Hình nó như hình 2.

- Cây quyết định (Decision Tree)

Cây quyết định (Decision Tree) là một cây phân cấp có cấu trúc được dùng để phân lớp các đối tượng dựa vào dãy các luật. Các thuộc tính của đối tượng có thể thuộc các kiểu dữ liệu khác nhau như Nhị phân (Binary), Định

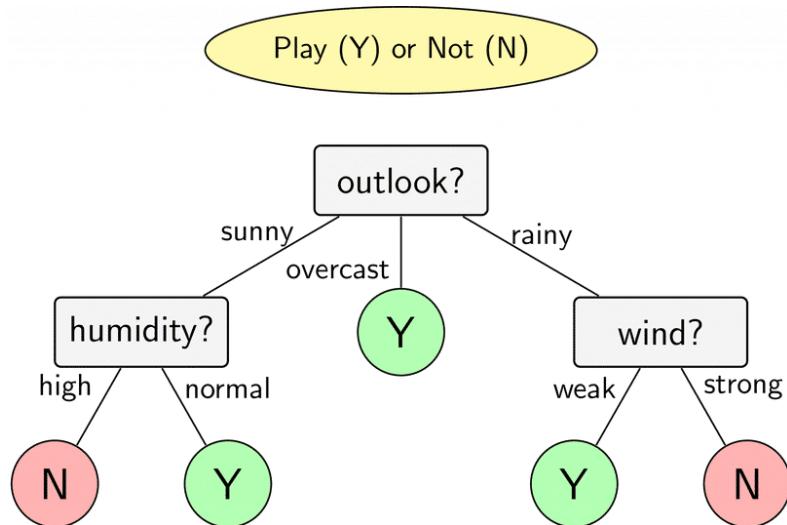
danh (Nominal), Thứ tự (Ordinal), Số lượng (Quantitative) trong khi đó thuộc tính phân lớp phải có kiểu dữ liệu là Binary hoặc Ordinal.

Tóm lại, cho dữ liệu về các đối tượng gồm các thuộc tính cùng với lớp (classes) của nó, cây quyết định sẽ sinh ra các luật để dự đoán lớp của các dữ liệu chưa biết.

Ta hãy xét một ví dụ 1 kinh điển khác về cây quyết định. Giả sử dựa theo thời tiết mà các bạn nam sẽ quyết định đi đá bóng hay không?

Những đặc điểm ban đầu là: Thời tiết, Độ ẩm, Gió.

Dựa vào những thông tin trên, bạn có thể xây dựng được mô hình như sau:



Hình 2.4: Mô hình cây quyết định.

Dựa theo mô hình trên, ta thấy:

Nếu trời nắng, độ ẩm bình thường thì khả năng các bạn nam đi chơi bóng sẽ cao. Còn nếu trời nắng, độ ẩm cao thì khả năng các bạn nam sẽ không đi chơi bóng.

- Thuật toán Cây quyết định (Decision Tree)
 - Thuật toán ID3.

Giờ chúng ta hãy cùng tìm hiểu cách thức hoạt động của thuật toán cây quyết định thông qua thuật toán đơn giản ID3.

ID3 (J. R. Quinlan 1993) sử dụng phương pháp tham lam tìm kiếm từ trên xuống thông qua không gian của các nhánh có thể không có backtracking. ID3 sử dụng Entropy và Information Gain để xây dựng một cây quyết định.

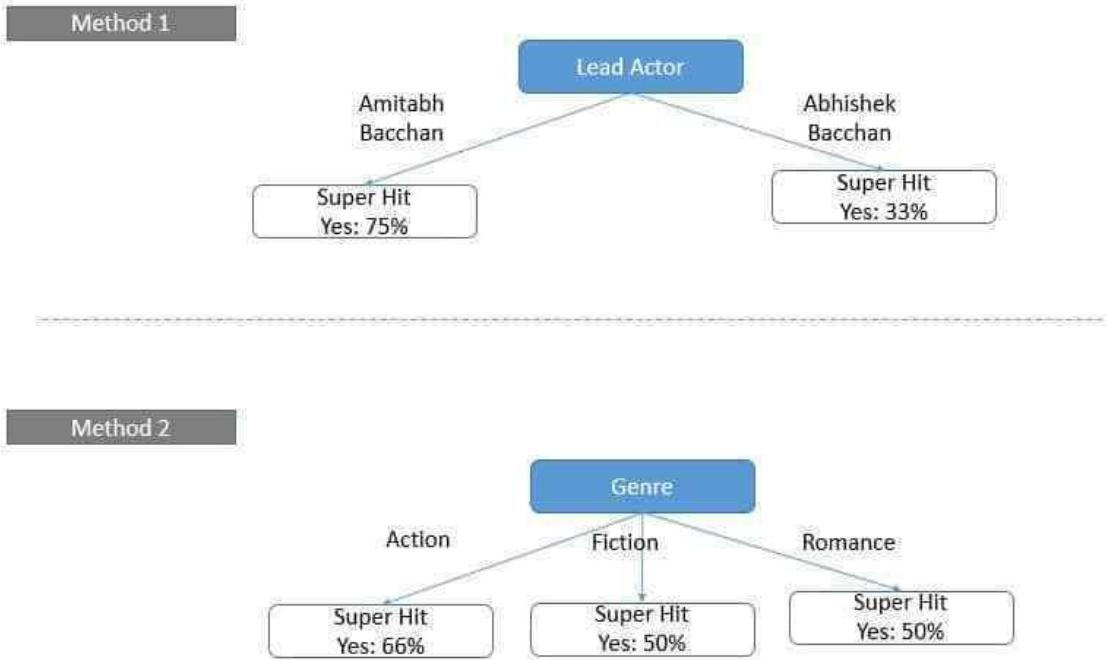
Ta xét ví dụ 2:

Bạn muốn xem xét sự thành công của một bộ phim thông qua hai yếu tố: diễn viên chính của phim và thể loại phim:

Bảng 2.1: Bảng mô tả dữ liệu

Lead Actor	Genre	Hit(Y/N)
Amitabh Bacchan	Action	Yes
Amitabh Bacchan	Fiction	Yes
Amitabh Bacchan	Romance	No
Amitabh Bacchan	Action	Yes
Abhishek Bacchan	Action	No
Abhishek Bacchan	Fiction	No
Abhishek Bacchan	Romance	Yes

Giả sử, bạn muốn xác định độ thành công của bộ phim chỉ trên 1 yếu tố, bạn sẽ có hai cách thực hiện sau: qua diễn viên chính của phim và qua thể loại phim.



Hình 2.5: Hai phương pháp xác định độ thành công.

Qua sơ đồ, ta có thể thấy rõ ràng ràng, với phương pháp thứ nhất, ta phân loại được rõ ràng, trong khi phương pháp thứ hai, ta có một kết quả lộn xộn hơn. Và tương tự, cây quyết định sẽ thực hiện như trên khi thực hiện việc chọn các biến.

Có rất nhiều hệ số khác nhau mà phương pháp cây quyết định sử dụng để phân chia. Dưới đây, tôi sẽ đưa ra hai hệ số phổ biến là Information Gain và Gain Ratio (ngoài ra còn hệ số Gini).

- Entropy trong Cây quyết định (Decision Tree)

Entropy là thuật ngữ thuộc Nhiệt động lực học, là thước đo của sự biến đổi, hỗn loạn hoặc ngẫu nhiên. Năm 1948, Shannon đã mở rộng khái niệm Entropy sang lĩnh vực nghiên cứu, thống kê với công thức như sau:

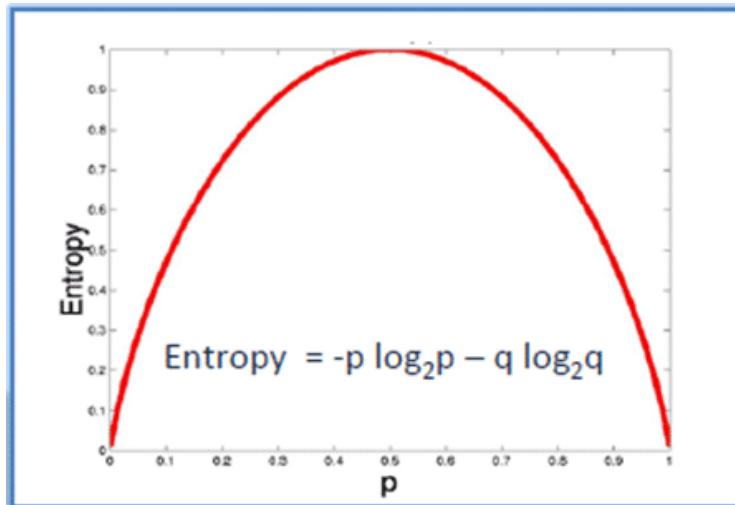
Với một phân phối xác suất của một biến rời rạc x có thể nhận n giá trị khác nhau x_1, x_2, \dots, x_n .

Giả sử rằng xác suất để x nhận các giá trị này là $p_i = p(x=x_i)$.

Ký hiệu phân phối này là $p=(p_1, p_2, \dots, p_n)$. Entropy của phân phối này được định nghĩa là: $H(p) = -\sum_{i=1}^n p_i \log(p_i)$

Giả sử bạn tung một đồng xu, entropy sẽ được tính như sau:

$$H = -[0.5 \ln(0.5) + 0.5 \ln(0.5)]$$



$$\text{Entropy} = -0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$$

Hình 2.6: Hàm Entropy.

Hình vẽ trên biểu diễn sự thay đổi của hàm entropy. Ta có thể thấy rằng, entropy đạt tối đa khi xác suất xảy ra của hai lớp bằng nhau.

P tinh khiết: $p_i = 0$ hoặc $p_i = 1$.

P vẫn đục: $p_i = 0.5$, khi đó hàm Entropy đạt đỉnh cao nhất.

- Information Gain trong Cây quyết định (Decision Tree).

Information Gain dựa trên sự giảm của hàm Entropy khi tập dữ liệu được phân chia trên một thuộc tính. Để xây dựng một cây quyết định, ta phải tìm tất cả thuộc tính trả về Infomation gain cao nhất.

Để xác định các nút trong mô hình cây quyết định, ta thực hiện tính Infomation Gain tại mỗi nút theo trình tự sau:

- Bước 1: Tính toán hệ số Entropy của biến mục tiêu S có N phần tử với Nc phần tử thuộc lớp c cho trước:

$$\circ \quad H(S) = - \sum_{c=1}^C (N_c/N) \log(N_c/N)$$

- Bước 2: Tính hàm số Entropy tại mỗi thuộc tính: với thuộc tính x, các điểm dữ liệu trong S được chia ra K child node S1, S2, ..., SK với số điểm trong mỗi child node lần lượt là m1, m2, ..., mK, ta có:

$$\circ \quad H(x, S) = \sum_{k=1}^K (m_k / N) * H(S_k)$$

- Bước 3: Chỉ số Gain Information được tính bằng:

$$\circ \quad G(x, S) = H(S) - H(x, S)$$

Với ví dụ 2 trên, ta tính được hệ số Entropy như sau:

$$\text{EntropyParent} = -(0.57 \ln(0.57) + 0.43 \ln(0.43)) = 0.68$$

Hệ số Entropy theo phương pháp chia thứ nhất:

$$\text{Entropyleft} = -(0.75 \ln(0.75) + 0.25 \ln(0.25)) = 0.56$$

$$\text{Entroptright} = -(0.33 \ln(0.33) + 0.67 \ln(0.67)) = 0.63$$

Ta có thể tính hệ số Information Gain như sau:

$$\text{Information Gain} = 0.68 - (4 \cdot 0.56 + 3 \cdot 0.63) / 7 = 0.09$$

Hệ số Entropy với phương pháp chia thứ hai như sau:

$$\text{Entropyleft} = -(0.67 \ln(0.67) + 0.33 \ln(0.33)) = 0.63$$

$$\text{Entropymiddle} = -(0.5 \ln(0.5) + 0.5 \ln(0.5)) = 0.69$$

$$\text{Entroptright} = -(0.5 \ln(0.5) + 0.5 \ln(0.5)) = 0.69$$

Hệ số Information Gain:

$$\text{Information Gain} = 0.68 - (3 \cdot 0.63 + 2 \cdot 0.69 + 2 \cdot 0.69) / 7 = 0.02$$

So sánh kết quả, ta thấy nếu chia theo phương pháp 1 thì ta được giá trị hệ số Information Gain lớn hơn gấp 4 lần so với phương pháp 2. Như vậy, giá trị thông tin ta thu được theo phương pháp 1 cũng nhiều hơn phương pháp 2.

- Thuật toán C4.5.

Thuật toán C4.5 là thuật toán cải tiến của ID3.

Trong thuật toán ID3, Information Gain được sử dụng làm độ đo. Tuy nhiên, phương pháp này lại ưu tiên những thuộc tính có số lượng lớn các giá trị mà ít xét tới những giá trị nhỏ hơn. Do vậy, để khắc phục nhược điểm trên, ta sử dụng độ đo Gain Ratio (trong thuật toán C4.5) như sau:

Đầu tiên, ta chuẩn hoá information gain với trị thông tin phân tách (split information):

$$\text{Gain Ratio} = \frac{\text{Information Gain}}{\text{Split Info}}$$

Trong đó: Split Info được tính như sau:

$$-\sum_{i=1}^n D_i \log_2 D_i$$

Giả sử chúng ta phân chia biến thành n nút cón và D_i đại diện cho số lượng bản ghi thuộc nút đó. Do đó, hệ số Gain Ratio sẽ xem xét được xu hướng phân phối khi chia cây.

Áp dụng cho ví dụ trên và với cách chia thứ nhất, ta có:

$$\text{Split Info} = -((4/7)*\log_2(4/7)) - ((3/7)*\log_2(3/7)) = 0.98$$

$$\text{Gain Ratio} = 0.09/0.98 = 0.092$$

Tiêu chuẩn dừng:

Trong các thuật toán Decision tree, với phương pháp chia trên, ta sẽ chia mãi các node nếu nó chưa tinh khiết. Như vậy, ta sẽ thu được một tree mà mọi điểm trong tập huấn luyện đều được dự đoán đúng (giả sử rằng không có hai input giống nhau nào cho output khác nhau). Khi đó, cây có thể sẽ rất phức tạp (nhiều node) với nhiều leaf node chỉ có một vài điểm dữ liệu. Như vậy, nhiều khả năng overfitting sẽ xảy ra.

Để tránh trường hợp này, ta có thể dùng cây theo một số phương pháp sau đây:

- Nếu node đó có entropy bằng 0, tức mọi điểm trong node đều thuộc một class.
- Nếu node đó có số phần tử nhỏ hơn một ngưỡng nào đó. Trong trường hợp này, ta chấp nhận có một số điểm bị phân lớp sai để tránh overfitting. Class cho leaf node này có thể được xác định dựa trên class chiếm đa số trong node.
- Nếu khoảng cách từ node đó đến root node đạt tới một giá trị nào đó. Việc hạn chế chiều sâu của tree này làm giảm độ phức tạp của tree và phần nào giúp tránh overfitting.
- Nếu tổng số leaf node vượt quá một ngưỡng nào đó.
- Nếu việc phân chia node đó không làm giảm entropy quá nhiều (information gain nhỏ hơn một ngưỡng nào đó).

Ngoài ra, ta còn có phương pháp cắt tỉa cây.

● Ưu / nhược điểm của thuật toán cây quyết định

- Ưu điểm:

Cây quyết định là một thuật toán đơn giản và phổ biến. Thuật toán này được sử dụng rộng rãi bởi những lợi ích của nó:

Mô hình sinh ra các quy tắc dễ hiểu cho người đọc, tạo ra bộ luật với mỗi nhánh lá là một luật của cây.

Dữ liệu đầu vào có thể là dữ liệu missing, không cần chuẩn hóa hoặc tạo biến giả.

Có thể làm việc với cả dữ liệu số và dữ liệu phân loại.

Có thể xác thực mô hình bằng cách sử dụng các kiểm tra thống kê.

Có khả năng làm việc với dữ liệu lớn.

- Nhược điểm:

Kèm với đó, cây quyết định cũng có những nhược điểm cụ thể:

Mô hình cây quyết định phụ thuộc rất lớn vào dữ liệu của bạn. Thậm chí, với một sự thay đổi nhỏ trong bộ dữ liệu, cấu trúc mô hình cây quyết định có thể thay đổi hoàn toàn.

Cây quyết định hay gặp vấn đề overfitting.

- Naive Bayes (Ngây thơ)

Naive Bayes là một thuật toán học máy phổ biến được sử dụng cho các nhiệm vụ phân loại. Thuật toán này dựa trên định lý Bayes và giả định "ngây thơ" (naive) về sự độc lập giữa các đặc trưng. Ý tưởng chính của Naive Bayes là tính toán xác suất của một mẫu dữ liệu thuộc vào từng lớp dựa trên các đặc trưng của mẫu đó. Giả định ngây thơ giả định rằng các đặc trưng độc lập với nhau, tức là sự xuất hiện hoặc vắng mặt của một đặc trưng không ảnh hưởng đến sự xuất hiện hoặc vắng mặt của các đặc trưng khác. Naive Bayes được áp dụng rộng rãi trong các vấn đề phân loại văn bản như phân tích cảm xúc, phân loại thư rác, phân loại văn bản tự nhiên, v.v. Nó cũng thường được sử dụng trong các bài toán phân loại hồi quy như dự đoán giá trị của một biến dự báo dựa trên các đặc trưng khác. Mặc dù Naive Bayes có giả định ngây thơ đơn giản, nhưng nó thường cho kết quả tốt và hoạt động hiệu quả trên các tập dữ liệu lớn. Đặc tính này làm cho Naive Bayes trở thành một trong những lựa chọn phổ biến trong học máy, đặc biệt là khi xử lý các tập dữ liệu lớn và có nhiều đặc trưng.

2.4 Kết luận chương 2

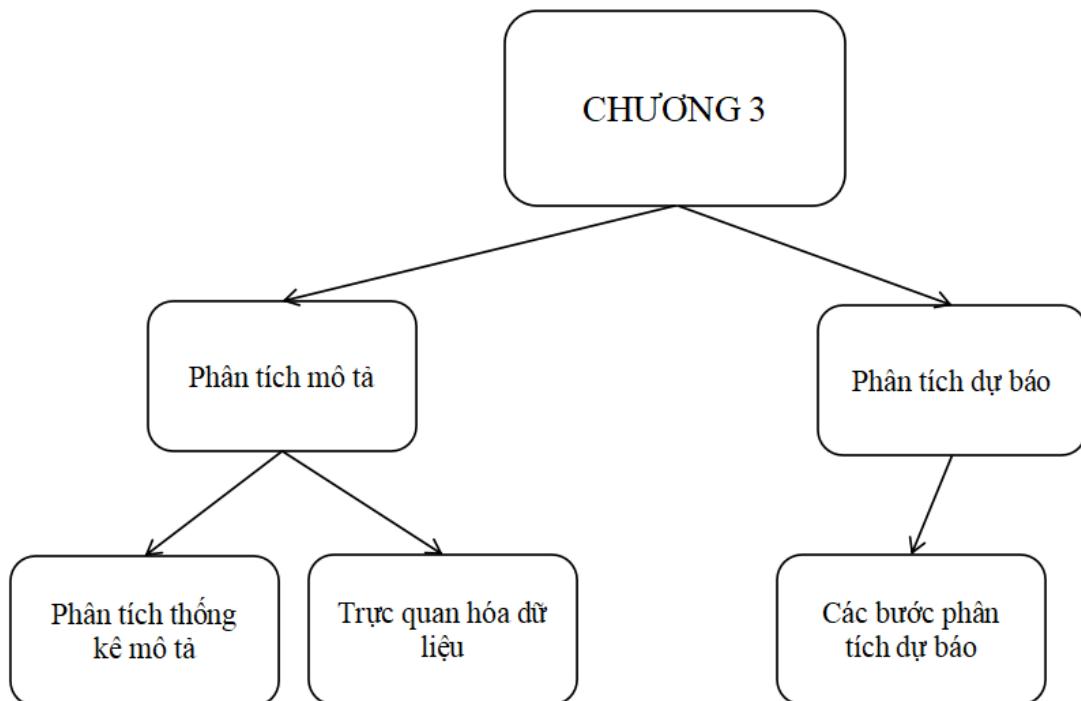
Chương 2 đã trình bày các phương pháp kỹ thuật, cụ thể là phương pháp phân tích mô tả, phương pháp phân tích hồi quy, phân loại và các công cụ thực hiện bài toán. Đồng thời lựa chọn được phương pháp phân tích như hồi quy tuyến tính, hồi quy Lasso, hồi quy Ridge, và ngôn ngữ Python để thực hiện thực nghiệm trên bài toán hồi quy.

Bên cạnh đó, đối với bài toán phân loại, ta cũng chọn lọc được 4 phương pháp phân tích quen thuộc và có hiệu quả khá cao để tiến hành thực nghiệm như phương pháp hồi quy Logistic, Support Vector Machine, Cây quyết định (Decision Tree), Naive Bayes (Ngây thơ). Ta sẽ tiến hành áp dụng những phương pháp trên bằng công cụ Python để có thể thấy được độ hiệu quả của mô hình qua những thông số đánh giá dành riêng cho từng bài toán.

CHƯƠNG 3 PHÂN TÍCH DỮ LIỆU BẰNG CÔNG CỤ PYTHON

3.1 Giới thiệu tổng quan chương 3

Trong chương 3, chúng ta sẽ sử dụng công cụ phổ biến nhất hiện nay để phân tích dữ liệu, đó chính là Python. Ta sẽ tiến hành phân tích mô tả và phân tích dự báo bằng python.



Hình 3.1: Sơ đồ tổng quát chương 3.

3.2 Phân tích mô tả bằng python

3.2.1 Phân tích thống kê mô tả bằng python

Phân tích mô tả là quá trình sử dụng các kỹ thuật thống kê và trực quan hóa dữ liệu để khám phá và hiểu rõ về tính chất của một tập dữ liệu. Mục tiêu chính của phân tích mô tả là cung cấp cái nhìn tổng quan về dữ liệu, từ đó giúp người phân tích đưa ra các nhận định, kỹ thuật, và quyết định có ý nghĩa. Ở đây, ta sẽ tiến hành phân tích thống kê để đo lường sự phân phôi của dữ liệu và

trực quan hóa dữ liệu bằng python để xem được các tần suất cũng như sự phụ thuộc của tập dữ liệu.

Đầu tiên sẽ tiến hành phân tích trên tập ‘Stock.csv’, dưới đây là tổng quan về bộ dữ liệu.

1	Date	Open	High	Low	Close	Adj Close	Volume
2	2014-03-18T00:00:00.000	18.782142639160156	18.99892807006836	18.757143020629883	18.9785709311035	16.716615676879883	209647200
3	2014-03-19T00:00:00.000	19.099286884093164	19.15142822265625	18.89285659790039	18.97357177734375	16.712121351623535	224750000
4	2014-03-20T00:00:00.000	19.02464256286621	19.02392959547266	18.8339296188983	18.882143020629883	16.63167953491211	208398400
5	2014-03-21T00:00:00.000	19.97749465942383	19.0625	18.797506010351562	19.031070709228516	16.76285934482242	374046400
6	2014-03-24T00:00:00.000	19.22986193847656	19.303571701049805	19.16928534541258	19.256786346435547	16.961671829223633	357008800
7	2014-03-25T00:00:00.000	19.33928688419922	19.491071791049885	19.2710748034668	19.4639282265625	17.14412307739578	282293200
8	2014-03-26T00:00:00.000	19.51871853637695	19.60714340289961	19.2450809392334	19.27785626782227	16.988022844672266	297678800
9	2014-03-27T00:00:00.000	19.286428435138886	19.33928688419922	19.11129121447754	19.19499969482422	16.98725135863222	222031600
10	2014-03-28T00:00:00.000	19.2257173798854	19.24785614913672	19.0803565979085	19.27375861293457	16.8883743281128	208564000
11	2014-03-31T00:00:00.000	19.2581240561521	19.31464385986328	19.148156061842773	19.169285727995273	16.884665407714844	168669200
12	2014-04-01T00:00:00.000	19.28512422094727	19.352508095627344	19.17835675048893	19.3446424639169193	16.099664473748633	200760000
13	2014-04-02T00:00:00.000	19.48999984741211	19.295808076299345	19.376785278320312	17.067371363846823	180428800	
14	2014-04-03T00:00:00.000	19.335376506015625	19.335376506015625	19.09545	19.34990899693297	16.162349461521	19.335376506015625
15	2014-04-04T00:00:00.000	19.27895614913672	19.303571319580078	19.9849852789121	19.9849852789121	16.7928466915011	275251200
16	2014-04-05T00:00:00.000	19.025572780961	19.060714304028996	19.0704809361527	19.060714304028996	19.0704809361527	209560400
17	2014-04-06T00:00:00.000	19.05270572780961	19.060714304028996	19.060714304028996	19.060714304028996	19.060714304028996	243883400
18	2014-04-07T00:00:00.000	19.66571423016916	19.94697162475586	19.64377118577695	19.940000574057617	16.68264198303222	206169600
19	2014-04-08T00:00:00.000	19.95270572780961	19.98857162475588	19.694642791748807	19.695714958561523	16.46747589111328	2196527000
20	2014-04-09T00:00:00.000	19.53571219580078	19.6725080610351562	19.4697859640983	19.557508061032334	16.345731735229492	271717600
21	2014-04-10T00:00:00.000	19.639286193847666	19.648571914404297	19.612758640923432	19.6314296722411	16.418048461755371	2867674000
22	2014-04-11T00:00:00.000	19.58171853637695	19.6151299991697666	19.56177857198461911	19.498571395874873	16.59387154953613	266498000
23	2014-04-12T00:00:00.000	19.591785278120312	19.610314256284559	19.536871129413475	19.536871129413475	19.536871129413475	214765600
24	2014-04-13T00:00:00.000	19.571428298950195	19.88457177734375	19.542856216438664	19.74785614913672	16.51339530448242	284334400
25	2014-04-14T00:00:00.000	19.85270572780961	19.9009991697666	19.712856216438664	19.97035789489746	16.799386825561523	182548800
26	2014-04-15T00:00:00.000	19.868621365356453	19.99392809361797	19.880357179104980	19.89286422729492	16.726053227391504	202563200
27	2014-04-16T00:00:00.000	19.895270572780961	19.973571897713475	19.802577179104980	19.8048177179104980	16.5074234060878906	394940000
28	2014-04-17T00:00:00.000	19.29331747973633	20.35714340289961	20.026671548461910	19.86673303226562	17.590911600	
29	2014-04-18T00:00:00.000	20.16178515732422	20.428213119568636	20.426429748535156	17.99199602789961	390275200	
30	2014-04-19T00:00:00.000	20.457147383569335	20.4482135772270588	21.276786861645588	16.68871629638672	669485600	
31	2014-04-20T00:00:00.000	21.28499992370663	21.284999924741211	21.05392837524414	21.15646210510254	16.033338928222656	337377600
32	2014-04-21T00:00:00.000	21.165714263916016	21.408214569991797	21.06428527830312	21.074642181396484	18.5628662109375	456646800
33	2014-04-22T00:00:00.000	21.14285659790039	21.233570098876953	21.096428527830312	21.096428527830312	21.096428527830312	244048000
34	2014-05-01T00:00:00.000	21.22142928428922	21.061071395874037	21.163570440482734	16.641113938992578	191514400	
35	2014-05-02T00:00:00.000	21.0764293676653	21.46428684019922	21.071428288950272461	19.4004817581176754	287067200	
36	2014-05-06T00:00:00.000	21.4022856979370117	21.586871914404297	21.2289295297685332	18.698743377685532	374564400	
37	2014-05-07T00:00:00.000	21.258928289950195	21.33178520826367	20.98035664552054	18.633338928222656	282854400	
38	2014-05-08T00:00:00.000	21.098828289950195	21.28482951965332	20.942856216438664	20.9996431235805664	18.60011863708499	232972700
39	2014-05-09T00:00:00.000	20.876428664125975	20.9375	20.91214370727539	18.522621154785156	291597600	
40	2014-05-12T00:00:00.000	20.981785278120312	21.2021427154541	21.073500010351562	18.752223419189453	213208800	
41	2014-05-13T00:00:00.000	21.14285659790039	21.233570098876953	21.096428527830312	18.782634735107422	150737200	
42	2014-05-14T00:00:00.000	21.15821456991797	21.33571434028996	21.133571245638664	18.76640242027812	166404000	
43	2014-05-15T00:00:00.000	21.23928422729492	21.0012428684125977	21.029285430986203	18.62636947631836	230846000	
44	2014-05-16T00:00:00.000	21.222499984427656	21.34035662768227	21.097142869160155	18.39643473893559	18.901268805371094	276220000
45	2014-05-17T00:00:00.000	21.351785659790039	21.690357208251953	21.332131866152344	18.25298566455058	19.12523078918457	317755200
46	2014-05-18T00:00:00.000	21.565357208251953	21.657142639160156	21.54564234951177	21.5678686499934318	18.1290283201215	234836000
47	2014-05-21T00:00:00.000	21.665357208251953	21.667856216430664	21.50214385986328	21.653928756713867	19.17964744567871	196859600
48	2014-05-22T00:00:00.000	21.664286565979064	21.700871363846823	21.575006762939453	21.688213348838672	21.204006713867188	200760000
49	2014-05-23T00:00:00.000	21.6675	21.954643124951172	21.659643173217773	19.42702293395996	232209600	

Hình 3.2: Dữ liệu về tập Stock.csv.

Khi đã đọc được dữ liệu bằng thư viện pandas của python, ta bắt đầu phân tích bằng cách xem thông tin của tập dữ liệu, kích cỡ, tên của những thuộc tính có trong bộ dữ liệu đó.

```
stock_data.shape
5]    ✓ 0.0s
·   (5035, 6)

stock_data.info()
'6]
·   <class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 5035 entries, 2003-10-20 to 2023-10-19
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   Open        5035 non-null   float64
 1   High         5035 non-null   float64
 2   Low          5035 non-null   float64
 3   Close        5035 non-null   float64
 4   Adj Close   5035 non-null   float64
 5   Volume       5035 non-null   int64  
dtypes: float64(5), int64(1)
memory usage: 275.4 KB

stock_data.keys()
# Những feature của tập dữ liệu
'7]
·   Index(['Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume'], dtype='object')
```

Hình 3.3: Tra cứu thông tin của tập dữ liệu.

Sử dụng thống kê mô tả (hàm describe) để tạo bảng thống kê mô tả về dữ liệu.

```

● stock_data.describe()
# Thống kê mô tả về tập dữ liệu
]



|       | Open        | High        | Low         | Close       | Adj Close   | Volume       |
|-------|-------------|-------------|-------------|-------------|-------------|--------------|
| count | 5035.000000 | 5035.000000 | 5035.000000 | 5035.000000 | 5035.000000 | 5.035000e+03 |
| mean  | 41.064532   | 41.517863   | 40.628309   | 41.090829   | 39.537489   | 4.090240e+08 |
| std   | 51.311743   | 51.904384   | 50.764408   | 51.358526   | 51.267895   | 3.956596e+08 |
| min   | 0.350893    | 0.355179    | 0.343750    | 0.351786    | 0.298202    | 3.145820e+07 |
| 25%   | 5.408036    | 5.499821    | 5.311250    | 5.435714    | 4.607745    | 1.151032e+08 |
| 50%   | 20.644644   | 20.922501   | 20.425714   | 20.718929   | 17.715302   | 2.754264e+08 |
| 75%   | 46.801250   | 47.198750   | 46.445000   | 46.817499   | 44.814857   | 5.759642e+08 |
| max   | 196.240005  | 198.229996  | 195.279999  | 196.449997  | 195.926956  | 3.372970e+09 |


```

Hình 3.4: Thống kê mô tả về tập dữ liệu.

Thay vì sử dụng hàm có sẵn, ta có thể thống kê dữ liệu như min, max, median cho các cột dữ liệu dạng số tùy ý. Dưới đây là ví dụ về thống kê cho cột ‘High’ bằng các câu lệnh thống kê độc lập.

```

> ^
    print('Min:',stock_data['High'].min())
    print('Max:',stock_data['High'].max())
    print('Mean:',stock_data['High'].mean())
    print('Median:',stock_data['High'].median())
    print('Mode:',stock_data['High'].mode().iloc[0])
    print('Count null:',stock_data['High'].isnull().sum())
    print('Q1, Q2, Q3 \n',stock_data['High'].quantile([0.25, 0.5, 0.75]))
    print('Var',stock_data['High'].var())
    print('STD',stock_data['High'].std())
[9]   ✓ 0.0s
...
Min: 0.3551790118217468
Max: 198.22999572753906
Mean: 41.5178631143158
Median: 20.922500610351562
Mode: 3.2142860889434814
Count null: 0
Q1, Q2, Q3
  0.25      5.499821
  0.50     20.922501
  0.75     47.198750
Name: High, dtype: float64
Var 2694.06512269908
STD 51.90438442654994

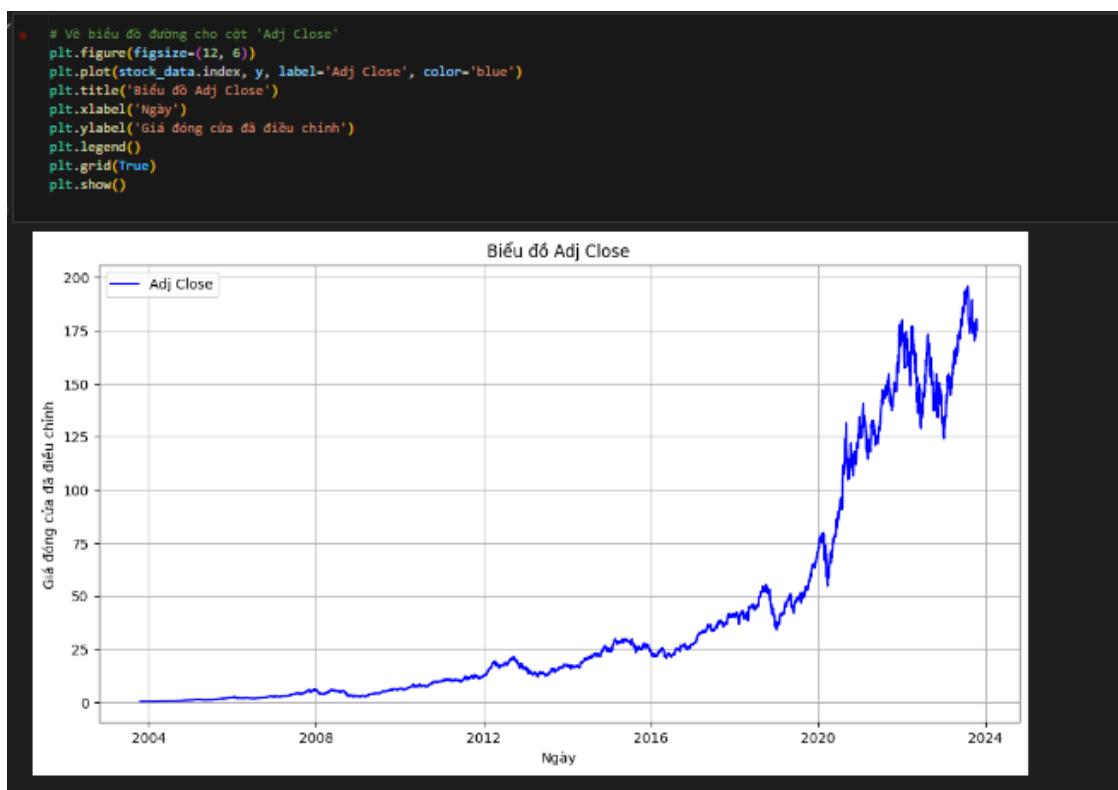
```

Hình 3.5: Thống kê mô tả về tập dữ liệu bằng câu lệnh độc lập.

3.2.2 Trực quan hóa dữ liệu bằng python

Trực quan hóa dữ liệu là quá trình quan trọng trong phân tích dữ liệu, nơi chúng ta chuyển đổi các dữ liệu số thành các biểu đồ và đồ thị để hiểu và trình bày thông tin một cách dễ hiểu và rõ ràng. Bằng cách sử dụng các biểu đồ và đồ thị, chúng ta có thể khám phá mối quan hệ, xu hướng và biến động trong dữ liệu một cách trực quan và mạch lạc. Việc trực quan hóa dữ liệu thường được áp dụng trên tập dữ liệu dạng số. Dưới đây là một số loại biểu đồ phổ biến thường được sử dụng trong trực quan hóa dữ liệu:

Biểu đồ đường (Line Chart): Biểu đồ đường được sử dụng để thể hiện sự thay đổi của một biến qua thời gian hoặc các giá trị liên tục khác. Đây là một công cụ mạnh mẽ để nhận diện xu hướng và biến động trong dữ liệu.



Hình 3.6: Trực quan hóa với biểu đồ đường.

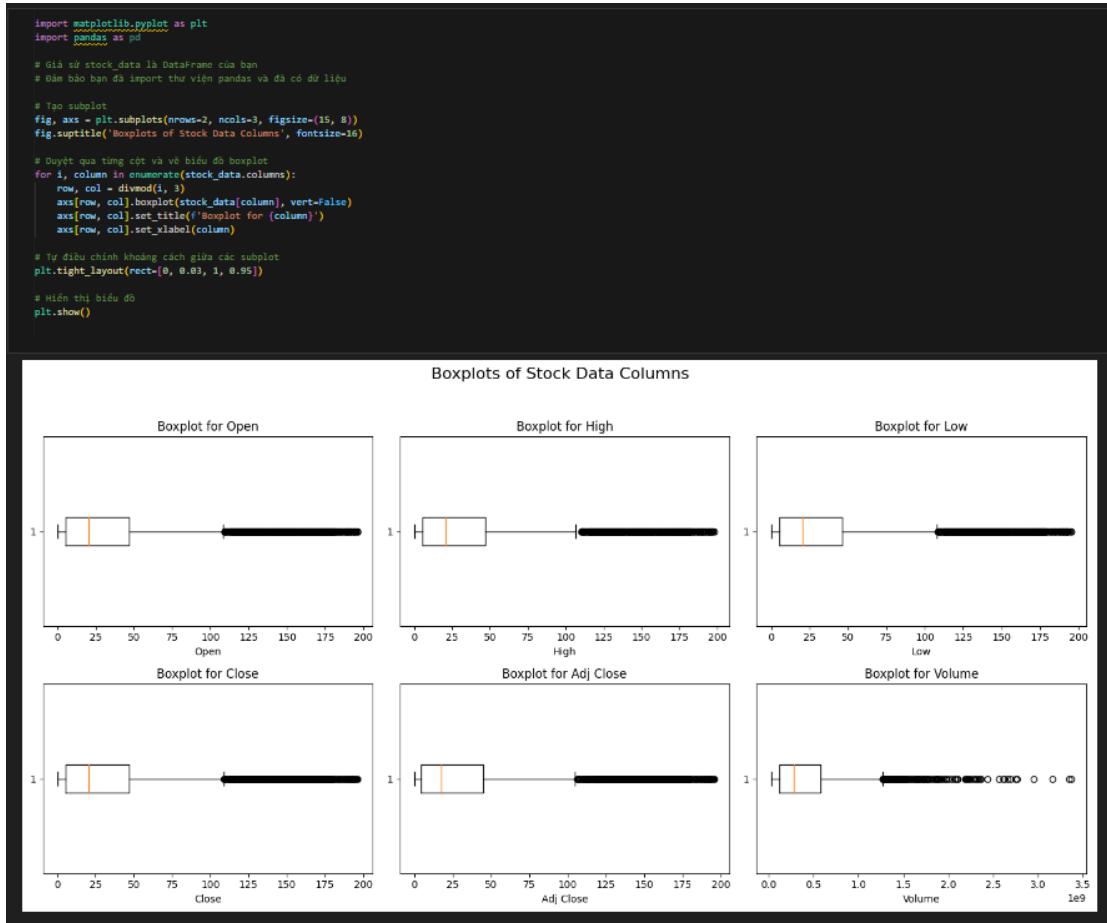
Biểu đồ cột (Bar Chart): Biểu đồ cột thường được sử dụng để so sánh các giá trị của các biến khác nhau hoặc của cùng một biến trong các nhóm khác

nhau. Đây là một công cụ hiệu quả để thể hiện sự khác biệt và xu hướng trong dữ liệu.



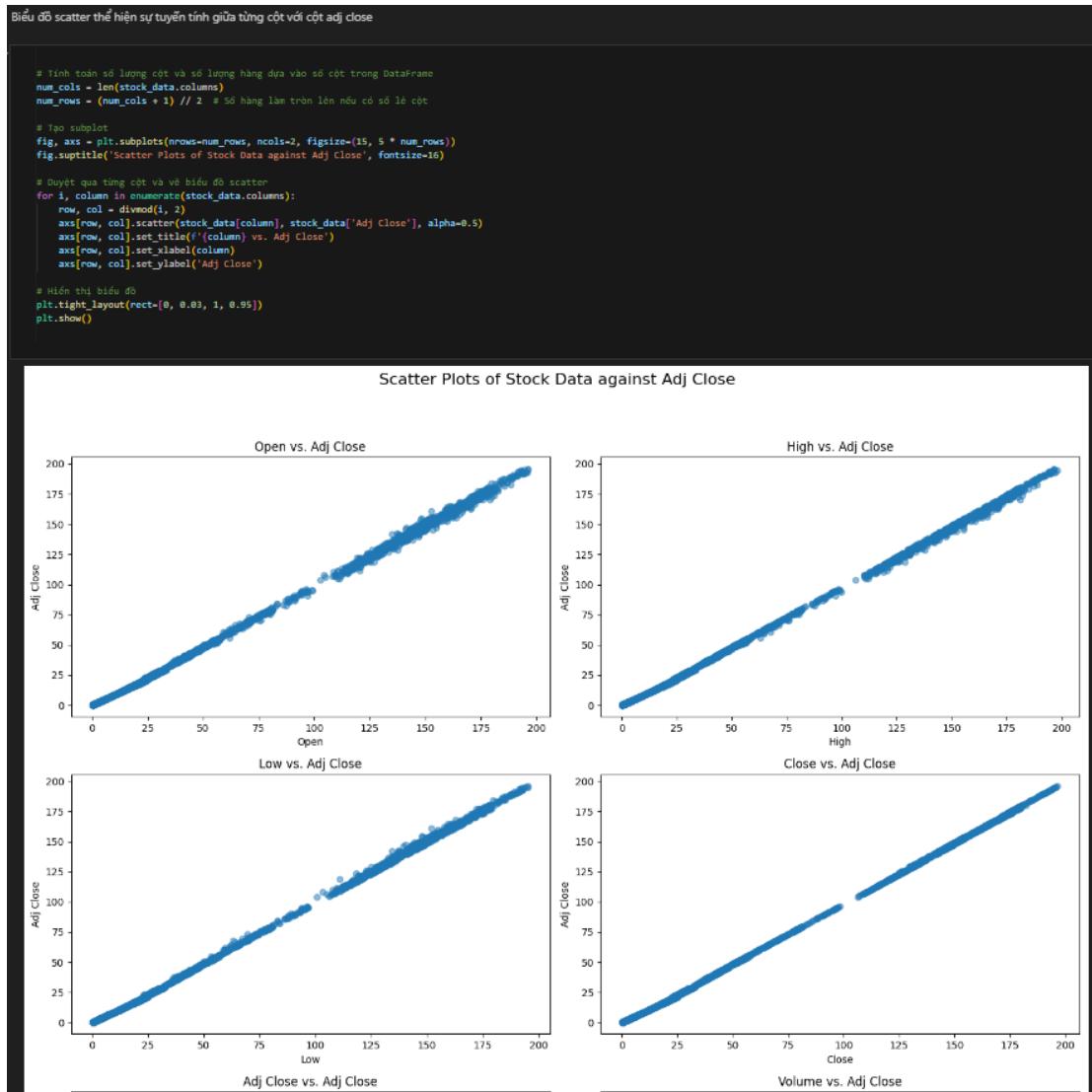
Hình 3.7: Trực quan hóa với biểu đồ cột.

Biểu đồ hộp và râu (Box and Whisker Plot): Biểu đồ hộp và râu được sử dụng để trực quan hóa phân phối và biến động của một biến. Nó cho thấy các giá trị trung vị, phân vị và phạm vi của dữ liệu một cách dễ dàng.



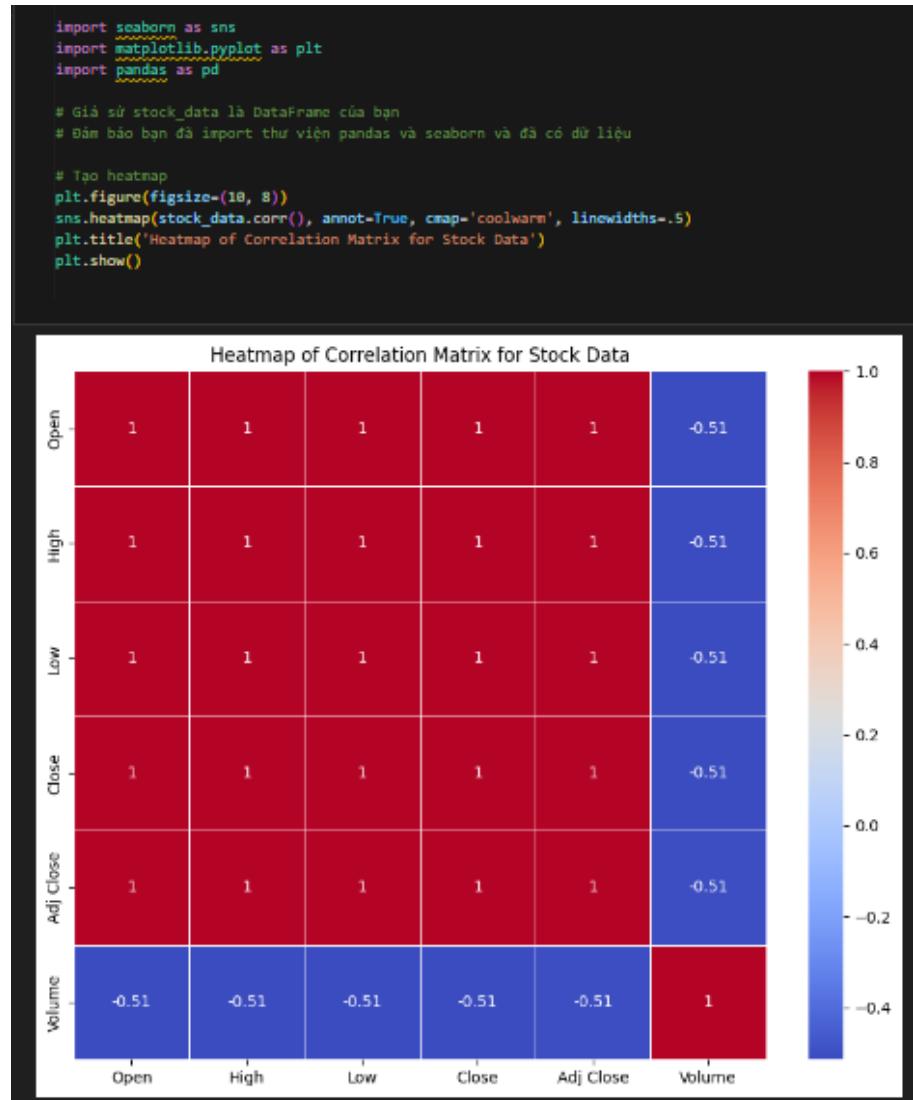
Hình 3.8: Trực quan hóa với biểu đồ hộp.

Biểu đồ phân tán (Scatter Plot): Biểu đồ phân tán được sử dụng để hiển thị mối quan hệ giữa hai biến. Nó cho phép chúng ta xác định mẫu và mối quan hệ tương quan giữa chúng.



Hình 3.9: Trục quan hóa với biểu đồ scatter.

Biểu đồ heatmap (Heatmap): Biểu đồ heatmap được sử dụng để hiển thị một ma trận bằng cách sử dụng các màu để biểu thị giá trị. Nó thường được sử dụng để hiển thị mật độ hoặc sự tương tác giữa các biến. Các loại biểu đồ này đều có ứng dụng rộng rãi trong phân tích dữ liệu và giúp chúng ta trình bày thông tin một cách dễ hiểu và hấp dẫn.



Hình 3.10: Trực quan hóa với biểu đồ heatmap.

3.3 Phân tích dự báo bằng python

Phân tích dự báo là quá trình nghiên cứu và đánh giá các dữ liệu lịch sử và thông tin hiện tại để tạo ra các ước lượng hoặc dự báo về tương lai. Mục tiêu của phân tích dự báo là cung cấp thông tin dự đoán chính xác nhất có thể về các sự kiện, xu hướng hoặc biến động tiềm ẩn trong tương lai. Các kỹ thuật phân tích dự báo thường dựa vào một loạt các mô hình thống kê, kỹ thuật máy học và các công cụ khác để phân tích dữ liệu và dự đoán kết quả.

Các bước chính trong quá trình phân tích dự báo bao gồm:

Thu thập dữ liệu: Thu thập dữ liệu lịch sử và dữ liệu hiện tại liên quan đến vấn đề hoặc hiện tượng cần dự báo. Ở đây, dữ liệu chúng ta sử dụng sẽ là bộ Stock.csv ở trên.

Tiền xử lý dữ liệu: Loại bỏ dữ liệu nhiễu, điều chỉnh giá trị thiếu, chuẩn hóa dữ liệu để chuẩn bị cho quá trình phân tích.

Đầu tiên, ta sẽ kiểm tra kiểu của dữ liệu, sau đó là kiểm tra khuyết và nhiễu nếu có.

```

[79] stock_data.shape
# Kích thước dữ liệu
...
... (5035, 6)

[80] stock_data.dtypes
#Kiểu dữ liệu của từng feature
...
... Open      float64
High       float64
Low        float64
Close      float64
Adj Close   float64
Volume     int64
dtype: object

[81] stock_data.isnull().sum()
#check missing value
...
... Open      0
High       0
Low        0
Close      0
Adj Close   0
Volume     0
dtype: int64

```

Ta thấy tập dữ liệu này khá là sạch

Hình 3.11: Tiền xử lý dữ liệu trước khi chia.

Sau khi đã có 1 tập dữ liệu sạch, ta tiến hành chia dữ liệu theo tập X_train, X_test, y_train, y_test theo biến độc lập X và biến tùy chọn y theo train_test_split.

```
# Chia dữ liệu thành tập huấn luyện và tập kiểm tra
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Hình 3.12: Chia dữ liệu.

Sau khi chia dữ liệu, ta sẽ chuẩn hóa tập dữ liệu X_train và sau đó chuyển đổi X_test theo X_train. Ngoài ra, ta có thể lưu mô hình chuẩn hóa để sử dụng cho việc chuẩn hóa dữ liệu mới sau này.

```
from sklearn.preprocessing import MinMaxScaler
minmax_scale = MinMaxScaler()
X_train = minmax_scale.fit_transform(X_train)
import pickle
with open("../Scaler_data/Scaler_Apple.pkl", "wb") as file:
    pickle.dump(minmax_scale, file)
X_test = minmax_scale.transform(X_test)

X_train, X_train.shape

(array([[0.11702975, 0.11667639, 0.11766868, 0.1168588 ],
       [0.01077164, 0.01074723, 0.01076149, 0.01081155],
       [0.12668891, 0.12754186, 0.12706   , 0.12697515],
       ...,
       [0.1201217 , 0.11954437, 0.11984881, 0.1190202 ],
       [0.2798301 , 0.27979721, 0.277953   , 0.28225034],
       [0.01531467, 0.015236   , 0.015365   , 0.01535358]]),
(4028, 4))

X_test,X_test.shape

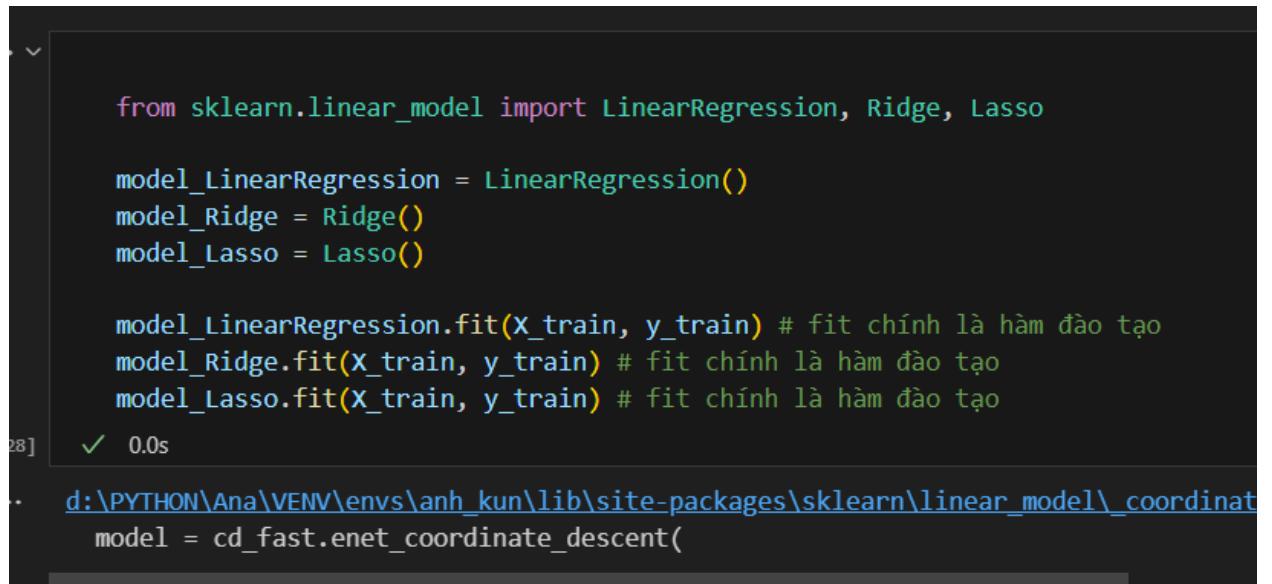
(array([[0.21525681, 0.21689127, 0.21733523, 0.21566912],
       [0.74988386, 0.74589996, 0.73368937, 0.72789433],
       [0.0981878 , 0.09711316, 0.09705425, 0.09666445],
       ...,
       [0.00663018, 0.00664291, 0.0065671 , 0.00656732],
       [0.87969482, 0.87679084, 0.88632392, 0.88576575],
       [0.01780977, 0.01819964, 0.01760225, 0.01784385]]),
(1007, 4))
```

Hình 3.13: Chuẩn hóa dữ liệu.

Lựa chọn mô hình: Chọn một hoặc nhiều mô hình phân tích phù hợp với loại dữ liệu và mục tiêu của dự báo. Với bài toán hồi quy, ta có các mô hình như Linear Regression, Ridge Regression, Lasso Regression. Còn với bài toán

phân loại, ta có các mô hình như Logistic Regression, SVM, Naive Bayes... Với bộ dữ liệu Stock.csv là bài toán hồi quy, ta tiến hành huấn luyện trên 3 mô hình Linear Regression, Ridge Regression, Lasso Regression.

Huấn luyện mô hình: Sử dụng dữ liệu lịch sử để huấn luyện mô hình, tức là điều chỉnh các thông số của mô hình để nó có thể dự đoán chính xác các giá trị trong tương lai. Ta sử dụng những mô hình đã chọn để tiến hành huấn luyện mô hình.



```

from sklearn.linear_model import LinearRegression, Ridge, Lasso

model_LinearRegression = LinearRegression()
model_Ridge = Ridge()
model_Lasso = Lasso()

model_LinearRegression.fit(X_train, y_train) # fit chính là hàm đào tạo
model_Ridge.fit(X_train, y_train) # fit chính là hàm đào tạo
model_Lasso.fit(X_train, y_train) # fit chính là hàm đào tạo
28]   ✓  0.0s
..  d:\PYTHON\Ana\VENV\envs\anh_kun\lib\site-packages\sklearn\linear_model\coordinat
      model = cd_fast.enet_coordinate_descent(

```

Hình 3.14: Huấn luyện mô hình.

Đánh giá mô hình: Đánh giá hiệu suất của mô hình bằng cách sử dụng các phương pháp kiểm định và đo lường hiệu suất. Đối với bài toán hồi quy, ta sử dụng các thông số như MAE, MSE, RMSE, R² để tiến hành đánh giá.

```

from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
from math import sqrt

✓ 0.0s

y_pred_linear = model_LinearRegression.predict(X_test)
y_pred_Lasso = model_Lasso.predict(X_test)
y_pred_Ridge = model_Ridge.predict(X_test)
def danh_gia(y_test, y_pred_linear,y_pred_Lasso,y_pred_Ridge):
    print('Kết quả trên hồi quy tuyến tính')
    mse_linear = mean_squared_error(y_pred_linear,y_test)
    mae_linear = mean_absolute_error(y_pred_linear,y_test)
    rmse_linear = sqrt(mean_squared_error(y_pred_linear,y_test))
    r2_linear = r2_score(y_pred_linear,y_test)
    print(f'MSE: {mse_linear}, MAE:{mae_linear}, RMSE:{rmse_linear}, R2:{r2_linear}')

    print('Kết quả trên hồi quy Lasso')
    mse_lasso = mean_squared_error(y_pred_Lasso,y_test)
    mae_lasso = mean_absolute_error(y_pred_Lasso,y_test)
    rmse_lasso = sqrt(mean_squared_error(y_pred_Lasso,y_test))
    r2_lasso = r2_score(y_pred_Lasso,y_test)
    print(f'MSE: {mse_lasso}, MAE:{mae_lasso}, RMSE:{rmse_lasso}, R2:{r2_lasso}')

    print('Kết quả trên hồi quy Ridge')
    mse_ridge = mean_squared_error(y_pred_Ridge,y_test)
    mae_ridge = mean_absolute_error(y_pred_Ridge,y_test)
    rmse_ridge = sqrt(mean_squared_error(y_pred_Ridge,y_test))
    r2_ridge = r2_score(y_pred_Ridge,y_test)
    print(f'MSE: {mse_ridge}, MAE:{mae_ridge}, RMSE:{rmse_ridge}, R2:{r2_ridge}')
danh_gia(y_test,y_pred_linear,y_pred_Lasso,y_pred_Ridge)
✓ 0.0s

Kết quả trên hồi quy tuyến tính
MSE: 0.8613680079224142, MAE:0.7977327176516295, RMSE:0.9280991369042502, R2:0.9996859134652113
Kết quả trên hồi quy Lasso
MSE: 17.03235460757272, MAE:3.007506688868841, RMSE:4.127027333029516, R2:0.9927294125966358
Kết quả trên hồi quy Ridge
MSE: 1.0929745894376541, MAE:0.8886976526518109, RMSE:1.04545425028437, R2:0.9996002863769687

```

Hình 3.15: Đánh giá mô hình.

Dự đoán và đánh giá: Sử dụng mô hình đã huấn luyện để dự đoán các giá trị trong tương lai và đánh giá độ chính xác của dự đoán. Ta sẽ nhập vào tập dữ liệu cho biến độc lập X, lúc này dữ liệu sẽ đi qua mô hình chuẩn hóa đã lưu trước đó, sau đó sẽ được huấn luyện bởi mô hình được chọn.

```

open_price = 147.080002
high_price = 147.949997
low_price = 142.529999
close_price = 142.639999

new_data_scaled = minmax_scale.transform([[open_price,high_price,low_price,close_price]])
predicted_linear = model_LinearRegression.predict(new_data_scaled)
predicted_lasso = model_Lasso.predict(new_data_scaled)
predicted_ridge = model_Ridge.predict(new_data_scaled)
print('Kết quả dự báo trên HQT: ',predicted_linear[0])
print('Kết quả dự báo trên Lasso: ',predicted_lasso[0])
print('Kết quả dự báo trên Ridge: ',predicted_ridge[0])

] ✓ 0.0s
Kết quả dự báo trên HQT: 140.78667841213962
Kết quả dự báo trên Lasso: 135.9398715523588
Kết quả dự báo trên Ridge: 142.92149576700714

```

Hình 3.16: Dự đoán trên tập dữ liệu mới.

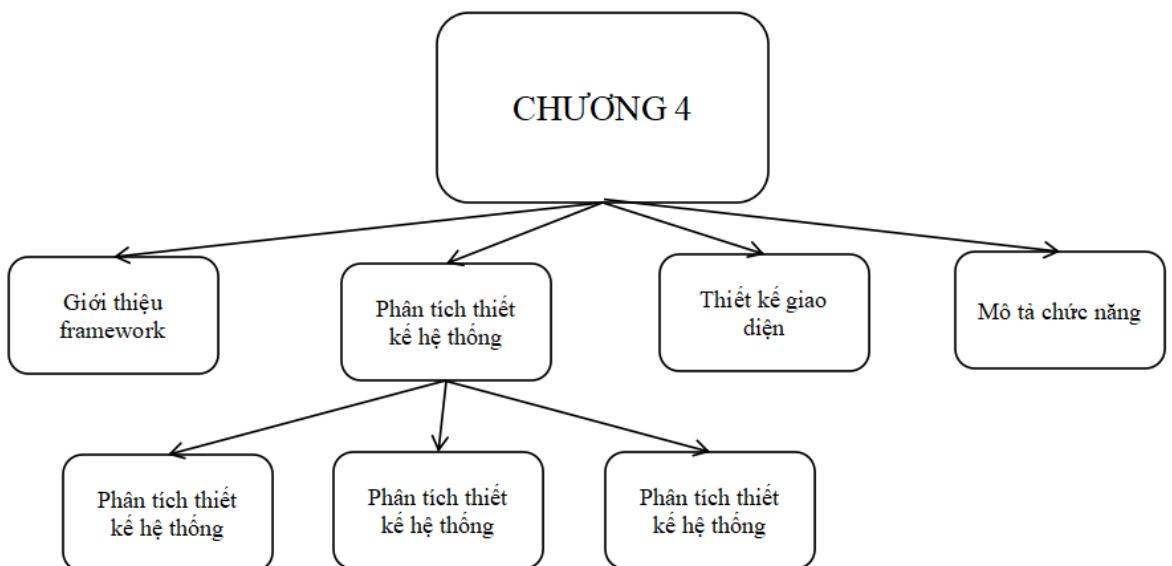
Triển khai và theo dõi: Áp dụng kết quả của dự báo vào thực tế và theo dõi hiệu suất của dự báo để có thể cải thiện và điều chỉnh trong tương lai.

Phân tích dự báo có thể được áp dụng trong nhiều lĩnh vực khác nhau như kinh tế, tài chính, marketing, y tế, và nhiều lĩnh vực khác để giúp các tổ chức và cá nhân ra quyết định và lập kế hoạch hiệu quả.

CHƯƠNG 4 XÂY DỰNG PHẦN MỀM HỖ TRỢ PHÂN TÍCH DỮ LIỆU

4.1 Giới thiệu tổng quan chương 4

Chương 4 là chương nói về cách thiết kế nền phần mềm ứng dụng, do vậy, chương này trước hết sẽ giới thiệu về framework được sử dụng. Sau đó sẽ tiến hành phân tích hệ thống bao gồm các mục lần lượt là: sơ đồ tổng quát, mô tả chi tiết use case, sau đó sẽ phân tích các use case đó. Tiếp theo sẽ tiến hành thiết kế hệ thống và mô tả các chức năng của hệ thống.



Hình 4.1: Sơ đồ tổng quát chương 4

4.2 Giới thiệu framework sử dụng

Giới thiệu về Streamlit

Streamlit là một thư viện Python mạnh mẽ được thiết kế để tạo các ứng dụng web tương tác một cách nhanh chóng và dễ dàng. Với Streamlit, việc xây dựng giao diện người dùng cho dữ liệu và mô hình không còn là thách thức lớn nữa. Thay vì tập trung vào lập trình HTML hoặc CSS, bạn có thể tận hưởng sự phát triển nhanh chóng bằng cách sử dụng các hàm Python đơn giản để hiển thị dữ liệu và tương tác với người dùng.



Hình 4.2: Logo streamlit.

Streamlit không chỉ đơn giản hóa quá trình phát triển ứng dụng web mà còn mang lại trải nghiệm linh hoạt cho người dùng cuối. Bạn có thể dễ dàng tích hợp biểu đồ, bảng và các thành phần tương tác khác vào ứng dụng của mình mà không cần kiến thức chuyên sâu về giao diện người dùng. Điều này làm cho Streamlit trở thành công cụ yêu thích của các nhà phân tích dữ liệu, nhà nghiên cứu và những người muốn nhanh chóng chia sẻ và triển khai các ứng dụng dựa trên Python một cách hiệu quả. Đồng thời, cộng đồng người dùng của Streamlit đang phát triển, cung cấp các nguồn lực và hỗ trợ đáng kể cho những người mới làm quen với công nghệ.

Khi nào thì nên sử dụng streamlit

Streamlit thật sự đã tạo ra những thuận lợi rất lớn cho cộng đồng machine learning, giúp việc xây dựng một web app trở nên dễ dàng hơn bao giờ hết trong khi chỉ mất cực ngắn để làm quen.

Mặc dù hầu hết các web app được xây dựng bởi Flask đều có thể transfer sang streamlit, framework này vẫn chưa thật sự hoàn thiện để thay thế được Flask. Tại sao lại vậy?

- Streamlit vẫn còn khá mới, và chưa thật sự đảm bảo được các vấn đề về bảo mật và an toàn cho hệ thống.
- Streamlit chưa hỗ trợ việc customize mạnh mẽ được như Flask (hoặc Django), hiện tại, nếu bạn muốn custom hoặc chỉnh sửa một method hiển thị

theo ý bạn, bạn cần clone lại source của streamlit và chỉnh sửa trực tiếp trên source.

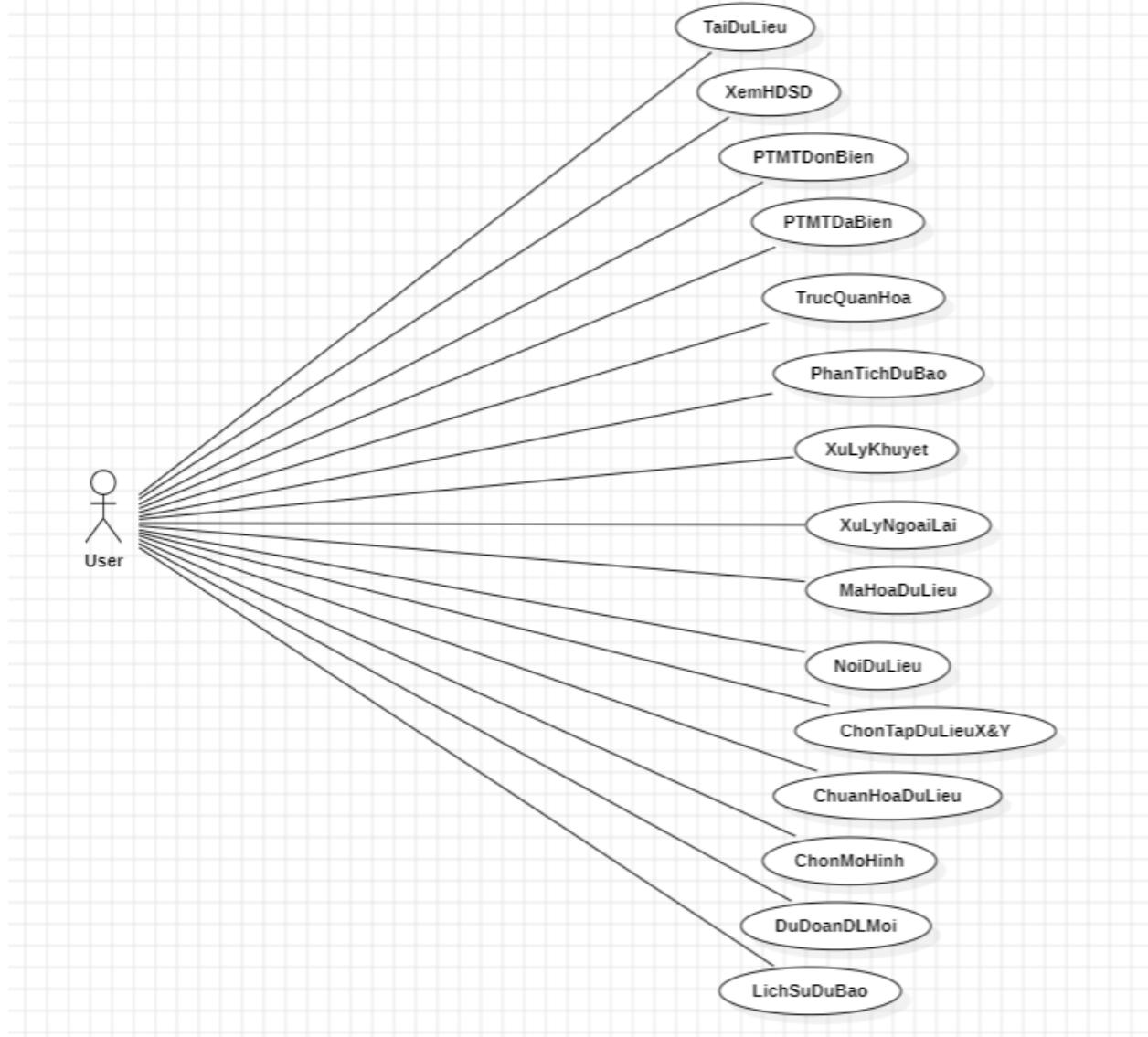
- Streamlit chưa hỗ trợ việc chuyển giao dữ liệu qua lại giữa các trang. Hiểu đơn giản là ví dụ bạn muốn truy cập vào trang xyz thì bạn cần vào trang login của xyz trước. Nhưng streamlit lại không đảm bảo được liên kết này.

Tóm lại, chỉ dùng streamlit:

- Cho dự án nhỏ, demo, PoC, ...
- Không yêu cầu quá cao về bảo mật
- Chỉ cần 1 trang dashboard duy nhất, không có quá nhiều trang liên kết đến nhau
- Chấp nhận layout mặc định của streamlit mà ko có nhu cầu custom quá nhiều.

4.3 Phân tích thiết kế hệ thống

4.3.1 Biểu đồ use case tổng quát



Hình 4.3: Biểu đồ use case tổng quát.

4.3.2 Mô tả chi tiết use case

1. Mô tả use case tải dữ liệu lên

Bảng 4.1: Use Case tải dữ liệu lên

Mã use case	UC1
-------------	-----

Tên use case	Tải dữ liệu lên
Tóm tắt	Use case này cho phép lựa chọn dữ liệu từ thiết bị để tải lên
Actor	Người dùng
Tiền điều kiện	Thiết bị của người dùng khởi động ứng dụng thành công
Đảm bảo tối thiểu	Người dùng không tải dữ liệu lên thì sẽ dừng lại ở trang chủ (Home)
Đảm bảo thành công	Người dùng tải được dữ liệu lên thành công và hiển thị tập dữ liệu lên mục ‘Home’
Kích hoạt	Người dùng ấn vào nút ‘Browse files’ và chọn tập dữ liệu từ thiết bị
Luồng sự kiện:	
<ol style="list-style-type: none"> 1. Người dùng truy cập vào trang chủ của ứng dụng 2. Nhấn nút ‘Browse files’ để tải lên dữ liệu từ thiết bị 3. Dữ liệu hiển thị lên màn hình, ở mục ‘Home’. 	

Ngoại lệ: Không có
Hậu điều kiện: Không có.

2 . Mô tả use case xem hướng dẫn sử dụng phần mềm

Bảng 4.2: use case xem hướng dẫn sử dụng phần mềm.

Mã use case	UC2
Tên use case	Xem hướng dẫn sử dụng phần mềm
Tóm tắt	Use case này cho phép người dùng xem hướng dẫn sử dụng phần mềm
Actor	Người dùng
Tiền điều kiện	Thiết bị của người dùng khởi động ứng dụng thành công
Đảm bảo tối thiểu	Người dùng truy cập được vào ứng dụng và thấy được mục ‘Hướng dẫn sử dụng phần mềm’
Đảm bảo thành công	Người dùng ấn vào mục ‘Hướng dẫn sử dụng phần mềm’ và hiển thị ra văn bản hướng dẫn
Kích hoạt	Người dùng ấn vào mục ‘Hướng dẫn sử dụng phần mềm’.

Luồng sự kiện:

1. Người dùng truy cập vào trang chủ của ứng dụng
2. Nhấn nút vào mục ‘Hướng dẫn sử dụng phần mềm’.
3. Văn bản hiển thị lên màn hình, ở mục ‘Home’.

Ngoại lệ: Không có**Hậu điều kiện:** Không có.**3. Mô tả use case phân tích mô tả đơn biến***Bảng 4.3: use case phân tích mô tả đơn biến.*

Mã use case	UC3
Tên use case	Phân tích mô tả đơn biến
Tóm tắt	Use case này cho phép người dùng xem thống kê mô tả bao gồm min, max,sum,mean,count null của 1 cột được chọn
Actor	Người dùng
Tiền điều kiện	Người dùng truy cập thành công vào trang ‘Phân tích mô tả’ sau khi upload dữ liệu thành công

Đảm bảo tối thiểu	Người dùng truy cập được vào giao diện phân tích mô tả. Khi người dùng chưa chọn 1 cột, sẽ mặc định là cột đầu tiên của tập dữ liệu
Đảm bảo thành công	Người dùng ánh chọn 1 cột dữ liệu bất kỳ, sau khi chọn sẽ hiển thị lên thống kê của cột đó
Kích hoạt	Người dùng ánh vào mũi tên xuống và chọn cột tùy ý.
Luồng sự kiện:	
1. Người dùng truy cập vào trang trang ‘Phân tích mô tả’ 2. Người dùng chọn 1 cột dữ liệu tùy ý sẽ hiển thị ra dữ liệu thống kê	
Ngoại lệ: Khi dữ liệu chưa được tải lên, giao diện của mục này sẽ không có gì cả.	
Hậu điều kiện: Không có.	

4. Mô tả use case phân tích mô tả đa biến

Bảng 4.4: use case phân tích mô tả đa biến.

Mã use case	UC4
Tên use case	Phân tích mô tả đa biến
Tóm tắt	Use case này cho phép người dùng xem thống kê mô tả bao gồm của 1 hoặc nhiều cột

Actor	Người dùng
Tiền điều kiện	Người dùng truy cập thành công vào trang ‘Phân tích mô tả’ sau khi upload dữ liệu thành công
Đảm bảo tối thiểu	Người dùng truy cập được vào giao diện phân tích mô tả. Người dùng thấy được mục ‘Analys Data’
Đảm bảo thành công	Người dùng tích chọn 1 loại tính năng phân tích bất kỳ, sau đó chọn các cột muốn phân tích. Sau khi chọn thành công, ấn nút ‘Analys’, dữ liệu sẽ hiển thị lên màn hình
Kích hoạt	Người dùng ấn nút ‘Analys’.
Luồng sự kiện:	
<ol style="list-style-type: none"> 1. Người dùng truy cập vào trang trang ‘Phân tích mô tả’ 2. Người dùng kích vào mục ‘Analys Data’ 3. Người dùng tích chọn 1 loại tính năng phân tích, chọn 1 hoặc nhiều cột dữ liệu, ấn nút ‘Analys’ 4. Dữ liệu được hiển thị ra màn hình 	
Ngoại lệ: Không có.	
Hậu điều kiện: Không có.	

5. Mô tả use case trực quan hóa dữ liệu

Bảng 4.5: use case trực quan hóa dữ liệu.

Mã use case	UC5
Tên use case	Trực quan hóa dữ liệu
Tóm tắt	Use case này cho phép người dùng xem 6 loại biểu đồ đường, cột, hộp, scatter, heatmap, tròn dựa trên dữ liệu của các cột đã chọn
Actor	Người dùng
Tiền điều kiện	Người dùng truy cập thành công vào trang ‘Phân tích mô tả’ sau khi upload dữ liệu thành công
Đảm bảo tối thiểu	Người dùng truy cập được vào giao diện phân tích mô tả. Người dùng thấy được mục ‘Trực quan hóa dữ liệu’
Đảm bảo thành công	Người dùng tích chọn 1 hoặc nhiều cột dữ liệu dữ liệu bất kỳ, các biểu đồ trên dashboard thay đổi theo những cột được chọn
Kích hoạt	Người dùng chọn 1 hoặc nhiều cột dữ liệu dữ liệu bất kỳ.
Luồng sự kiện:	
1. Người dùng truy cập vào trang trang ‘Phân tích mô tả’	

- | |
|---|
| <p>2. Người dùng kích vào mục ‘Trực quan hóa dữ liệu’</p> <p>3. Người dùng tích chọn 1 hoặc nhiều cột dữ liệu.</p> <p>4. Biểu đồ thay đổi được hiển thị ra màn hình</p> |
|---|

Ngoại lệ: Không có.

Hậu điều kiện: Không có.

6. Mô tả use case phân tích dự báo

Bảng 4.6: use case phân tích dự báo.

Mã use case	UC6
Tên use case	Phân tích dự báo
Tóm tắt	Use case này cho phép người dùng chuyển hướng đến giao diện phân tích dự báo và hiển thị lên dữ liệu đã upload
Actor	Người dùng
Tiền điều kiện	Người dùng đã upload dữ liệu lên thành công
Đảm bảo tối thiểu	Người dùng truy cập được vào giao diện phân tích dự báo

Đảm bảo thành công	Người dùng ấn vào mục ‘Phân tích dự báo’ và hiển thị lên dữ liệu đã upload trên giao diện đó
Kích hoạt	Người dùng ấn vào mục ‘Phân tích dự báo’.
Luồng sự kiện:	
1. Người dùng truy cập vào trang chủ của ứng dụng 2. Nhấn nút vào mục ‘Phân tích dự báo’. 3. Màn hình hiển thị dữ liệu đã upload 4. - Trên dataframe được hiển thị có biểu tượng download, người dùng có thể download tập dữ liệu về máy khi click vào biểu tượng này. - Trên dataframe được hiển thị có biểu tượng tìm kiếm, người dùng có thể tìm kiếm thông tin trong tập dữ liệu được hiển thị	
Ngoại lệ: Khi dữ liệu chưa được tải lên, giao diện của mục này sẽ không có gì cả.	
Hậu điều kiện: Không có.	

7. Mô tả use case xử lý giá trị khuyết

Bảng 4.7: use case xử lý giá trị khuyết.

Mã use case	UC7
Tên use case	Xử lý giá trị khuyết

Tóm tắt	Use case này cho phép người dùng xử lý các giá trị khuyết của tập dữ liệu dạng số bằng phương pháp điền khuyết
Actor	Người dùng
Tiền điều kiện	Người dùng đã truy cập thành công vào được trang ‘Phân tích dự báo’ sau khi upload dữ liệu thành công
Đảm bảo tối thiểu	Người dùng truy cập được vào giao diện phân tích dự báo, thấy được mục ‘Xử lý giá trị khuyết’
Đảm bảo thành công	Người dùng ấn vào mục ‘Xử lý giá trị khuyết’, sau đó chọn phương pháp điền khuyết tùy ý rồi ấn nút ‘Xử lý khuyết’, dữ liệu sau khi xử lý khuyết sẽ được hiển thị lên màn hình
Kích hoạt	Người dùng ấn nút ‘Xử lý khuyết’.
Luồng sự kiện:	
1. Người dùng truy cập vào trang phân tích dự báo 2. Nhấn nút vào mục ‘Xử lý giá trị khuyết’.	

3. Người dùng chọn 1 trong 3 phương pháp điền khuyết
4. Người dùng ấn nút ‘Xử lý khuyết’
5. Hệ thống sẽ hiển thị dữ liệu sau khi điền khuyết lên màn hình.
- Trên dữ liệu được hiển thị có biểu tượng download, người dùng có thể download tập dữ liệu về máy khi click vào biểu tượng này.
 - Trên dữ liệu được hiển thị có biểu tượng tìm kiếm, người dùng có thể tìm kiếm thông tin trong tập dữ liệu được hiển thị.

Ngoại lệ: Không có

Hậu điều kiện: Không có.

8. Mô tả use case xử lý ngoại lai

Bảng 4.8: use case xử lý ngoại lai.

Mã use case	UC8
Tên use case	Xử lý giá trị ngoại lai
Tóm tắt	Use case này cho phép người dùng xử lý các giá trị ngoại lai của 1 cột dữ liệu tùy chọn
Actor	Người dùng
Tiền điều kiện	Người dùng đã truy cập thành công vào được trang ‘Phân tích dự báo’ sau khi upload dữ liệu thành công

Đảm bảo tối thiểu	Người dùng truy cập được vào giao diện phân tích dự báo, thấy được mục ‘Xử lý giá trị ngoại lai’
Đảm bảo thành công	Người dùng ấn vào mục ‘Xử lý giá trị ngoại lai’, sau đó chọn 1 cột dữ liệu bất kỳ để xử lý. Lúc này, sẽ hiện ra 2 mục “Giá trị nhỏ nhất” và “Giá trị lớn nhất” để người dùng nhập vào. Khi nhập xong, ấn nút ‘Thực hiện xử lý’, dữ liệu sẽ hiển thị lên màn hình
Kích hoạt	Người dùng ấn nút “Thực hiện xử lý”

Luồng sự kiện:

1. Người dùng truy cập vào trang phân tích dự báo
2. Nhấn nút vào mục ‘Xử lý giá trị ngoại lai’.
3. Người dùng chọn 1 cột dữ liệu bất kỳ
4. Người dùng điền “Giá trị nhỏ nhất” và “Giá trị lớn nhất”
5. Người dùng ấn nút “Thực hiện xử lý”.
6. Dữ liệu sau khi xử lý được hiển thị lên màn hình.
 - Trên dữ liệu được hiển thị có biểu tượng download, người dùng có thể download tập dữ liệu về máy khi click vào biểu tượng này.
 - Trên dữ liệu được hiển thị có biểu tượng tìm kiếm, người dùng có thể tìm kiếm thông tin trong tập dữ liệu được hiển thị.

Ngoại lệ: Nếu trước đó có bước “Xử lý khuyết” thì sẽ thực hiện tiếp trên tập dữ liệu đã xử lý khuyết.

- Nếu trước đó không có bước ‘Xử lý khuyết’ thì sẽ thực hiện trên tập dữ liệu upload ban đầu (dạng số)

Hậu điều kiện: Không có.

9. Mô tả use case mã hóa dữ liệu

Bảng 4.9: use case mã hóa dữ liệu.

Mã use case	UC9
Tên use case	Mã hóa dữ liệu
Tóm tắt	Use case này cho phép người dùng mã hóa cột dữ liệu dạng chữ tùy chọn
Actor	Người dùng
Tiền điều kiện	Người dùng đã truy cập thành công vào được trang ‘Phân tích dự báo’ sau khi upload dữ liệu thành công
Đảm bảo tối thiểu	Người dùng truy cập được vào giao diện phân tích dự báo, thấy được mục ‘Mã hóa dữ liệu dạng chữ’
Đảm bảo thành công	Người dùng ấn vào mục ‘Mã hóa dữ liệu dạng chữ’, sau đó chọn 1 cột dữ liệu dạng chữ bất kỳ để xử lý. Sau

	đó người dùng ấn nút “Mã hóa”, dữ liệu dạng chữ sẽ hiển thị lên màn hình
Kích hoạt	Người dùng ấn nút “Mã hóa”.
Luồng sự kiện:	
<ol style="list-style-type: none"> 1. Người dùng truy cập vào trang phân tích dự báo 2. Nhấn nút vào mục ‘Mã hóa dữ liệu dạng chữ’. 3. Người dùng chọn 1 cột dữ liệu dạng chữ bất kỳ 4. Người dùng ấn nút “Mã hóa”. 5. Dữ liệu sau khi mã hóa được hiển thị lên màn hình <ul style="list-style-type: none"> - Trên dữ liệu được hiển thị có biểu tượng download, người dùng có thể download tập dữ liệu về máy khi click vào biểu tượng này. - Trên dữ liệu được hiển thị có biểu tượng tìm kiếm, người dùng có thể tìm kiếm thông tin trong tập dữ liệu được hiển thị. 	
Ngoại lệ: Nếu không có dữ liệu dạng chữ trong tập dữ liệu thì sẽ không thể chọn cột dữ liệu trong phần này	
Hậu điều kiện: Không có.	

10. Mô tả use case nối dữ liệu

Bảng 4.10: use case nối dữ liệu.

Mã use case	UC10
Tên use case	Nối dữ liệu

Tóm tắt	Use case này cho phép người dùng hợp nhất tập dữ liệu dạng số và chữ sau khi tiền xử lý
Actor	Người dùng
Tiền điều kiện	Người dùng đã truy cập thành công vào được trang ‘Phân tích dự báo’ sau khi upload dữ liệu thành công
Đảm bảo tối thiểu	Người dùng truy cập được vào giao diện phân tích dự báo, thấy được nút “Nối 2 loại dữ liệu”
Đảm bảo thành công	Người dùng ấn nút “Nối 2 loại dữ liệu” sẽ hiển thị lên data frame sau khi hợp nhất lên màn hình
Kích hoạt	Người dùng ấn nút “Nối 2 loại dữ liệu”

Luồng sự kiện:

1. Người dùng truy cập vào trang phân tích dự báo
2. Nhấn nút “Nối 2 loại dữ liệu”.
3. Hệ thống hiển thị dữ liệu đã hợp nhất trên màn hình.
 - Trên dữ liệu được hiển thị có biểu tượng download, người dùng có thể download tập dữ liệu về máy khi click vào biểu tượng này.
 - Trên dữ liệu được hiển thị có biểu tượng tìm kiếm, người dùng có thể tìm kiếm thông tin trong tập dữ liệu được hiển thị.

Ngoại lệ: Nếu trước đó có các bước “Xử lý khuyết”, “Xử lý ngoại lai”, “Mã hóa dữ liệu dạng chữ” thì dataframe sau khi nối sẽ gồm các dữ liệu sau khi xử lý từng bước.

- Nếu các bước trên chưa được thực hiện, data frame sau khi nối sẽ là data frame ban đầu được upload

Hậu điều kiện: Không có.

11. Mô tả use case chọn tập X và Y

Bảng 4.11: use case chọn tập X và Y.

Mã use case	UC11
Tên use case	Chọn tập dữ liệu X và Y
Tóm tắt	Use case này cho phép người dùng chọn biến độc lập X và biến phụ thuộc Y
Actor	Người dùng
Tiền điều kiện	Người dùng đã truy cập thành công vào được trang ‘Phân tích dự báo’ sau khi upload dữ liệu thành công
Đảm bảo tối thiểu	Người dùng truy cập được vào giao diện phân tích dự báo, thấy được mục “Chọn tập X và Y”

Đảm bảo thành công	Người dùng chọn 1 hoặc nhiều cột dữ liệu cho biến X và 1 cột dữ liệu cho biến Y
Kích hoạt	Người dùng ấn nút “Chia tập dữ liệu”
Luồng sự kiện:	
<ol style="list-style-type: none"> 1. Người dùng truy cập vào trang phân tích dự báo 2. Nhấn và mục “chọn tập X và Y” 3. Người dùng chọn các cột dữ liệu cho biến X và Y. 4. Người dùng ấn nút “Chia tập dữ liệu” 5. Hệ thống hiển thị tập dữ liệu đã chia lên màn hình bao gồm X, X_train, X_test, Y, Y_train, Y_test. <ul style="list-style-type: none"> - Trên dữ liệu được hiển thị có biểu tượng download, người dùng có thể download tập dữ liệu về máy khi click vào biểu tượng này. - Trên dữ liệu được hiển thị có biểu tượng tìm kiếm, người dùng có thể tìm kiếm thông tin trong tập dữ liệu được hiển thị. 	
Ngoại lệ: Nếu trước đó có các bước tiền xử lý thì sẽ áp dụng trên tập dữ liệu đã được xử lý trước đó.	
Hậu điều kiện: Không có.	

12. Mô tả use case chuẩn hóa dữ liệu

Bảng 4.12: use case chuẩn hóa dữ liệu.

Mã use case	UC12
Tên use case	Chuẩn hóa dữ liệu

Tóm tắt	Use case này cho phép người dùng chuẩn hóa tập X_train, sau đó chuyển đổi X_test theo X_train đã chuẩn hóa
Actor	Người dùng
Tiền điều kiện	Người dùng đã thực hiện thành công bước “Chọn tập dữ liệu X và Y”
Đảm bảo tối thiểu	Người dùng truy cập được vào giao diện phân tích dự báo, thấy được mục “Chuẩn hóa dữ liệu”
Đảm bảo thành công	Người dùng chọn 1 phương pháp chuẩn hóa, sau đó ấn nút “Chuẩn hóa”, dữ liệu X_train và X_test sau khi chuẩn hóa sẽ hiển thị lên màn hình
Kích hoạt	Người dùng ấn nút “Chuẩn hóa”
Luồng sự kiện:	
1. Người dùng truy cập vào trang phân tích dự báo 2. Nhấn và mục “Chuẩn hóa dữ liệu” 3. Người dùng chọn 1 phương pháp chuẩn hóa tùy ý. 4. Người dùng ấn nút “Chuẩn hóa” 5. Hệ thống hiển thị tập dữ liệu đã chuẩn hóa lên màn hình.	
- Trên dữ liệu được hiển thị có biểu tượng download, người dùng có thể download tập dữ liệu về máy khi click vào biểu tượng này. - Trên dữ liệu được hiển thị có biểu tượng tìm kiếm, người dùng có thể tìm kiếm thông tin trong tập dữ liệu được hiển thị.	

Ngoại lệ: Không có
Hậu điều kiện: Không có.

13. Mô tả use case chọn mô hình

Bảng 4.13: use case chọn mô hình.

Mã use case	UC13
Tên use case	Chọn mô hình
Tóm tắt	Use case này cho phép người dùng chọn mô hình và thông số đánh giá phù hợp với mô hình đó
Actor	Người dùng
Tiền điều kiện	Người dùng đã thực hiện thành công bước “Chuẩn hóa dữ liệu”
Đảm bảo tối thiểu	Người dùng truy cập được vào giao diện phân tích dự báo, thấy được mục “Chọn mô hình”
Đảm bảo thành công	Người dùng chọn 1 mô hình thuộc loại hồi quy hoặc phân loại, sau khi chọn xong mô hình sẽ hiện lên phần chọn thông số đánh giá phù hợp, chọn xong thông số, ấn nút “Huấn luyện” sẽ hiển thị thông số đánh giá lên màn hình.

Kích hoạt	Người dùng ấn nút “Huấn luyện”
Luồng sự kiện:	
<ol style="list-style-type: none"> 1. Người dùng truy cập vào trang phân tích dự báo 2. Nhấn và mục “Chọn mô hình” 3. Người dùng chọn 1 mô hình thuộc loại hồi quy hoặc phân loại để phù hợp với bài toán. 3. Sau khi đã chọn mô hình sẽ hiển thị lên mục “Chọn thông số đánh giá” 4. Người dùng chọn thông số đánh giá 5. Người dùng ấn nút “Huấn luyện” 6. Hệ thống hiển thị kết quả đánh giá lên màn hình. 	
Ngoại lệ: Không có	
Hậu điều kiện: Không có.	

14. Mô tả use case dự đoán dữ liệu mới

Bảng 4.14: use case dự đoán dữ liệu mới.

Mã use case	UC14
Tên use case	Dự đoán dữ liệu mới
Tóm tắt	Use case này cho phép người dùng dự đoán kết quả dựa trên tập dữ liệu họ nhập vào
Actor	Người dùng

Tiền điều kiện	Người dùng đã thực hiện thành công bước “Chọn mô hình”
Đảm bảo tối thiểu	Người dùng truy cập được vào giao diện phân tích dự báo, thấy được mục “Dự đoán với dữ liệu mới”
Đảm bảo thành công	Người dùng nhập dữ liệu cho các cột, sau đó ấn nút “Dự báo” sẽ hiển thị kết quả dự báo lên màn hình
Kích hoạt	Người dùng ấn nút “Dự báo”
Luồng sự kiện:	
<ol style="list-style-type: none"> 1. Người dùng truy cập vào trang phân tích dự báo 2. Nhấn và mục “Dự đoán với dữ liệu mới”, 3. Người dùng nhập dữ liệu cho từng cột. 4. Người dùng chọn ấn nút “Dự báo” 5. Hệ thống hiển thị kết quả sau khi dự báo lên màn hình. 	
Ngoại lệ: Không có	
Hậu điều kiện: Không có.	

15. Mô tả use case lịch sử dự báo

Bảng 4.15: use case lịch sử dự báo.

Mã use case	UC15
Tên use case	Lịch sử dự báo
Tóm tắt	Use case này cho phép người dùng xem lại lịch sử của các lần dự báo trước đó và tải xuống dữ liệu lịch sử
Actor	Người dùng
Tiền điều kiện	Người dùng đã thực hiện thành công bước “Dự báo trên dữ liệu mới”
Đảm bảo tối thiểu	Người dùng truy cập được vào giao diện “Lịch sử dự báo”
Đảm bảo thành công	Người dùng xem được lịch sử của các lần dự báo trước đó và tải được dữ liệu dự báo
Kích hoạt	Người dùng ấn chọn mục “Lịch sử dự báo”
Luồng sự kiện:	
1. Người dùng truy cập vào trang “Lịch sử dự báo” 2. Dữ liệu dự báo được hiển thị lên màn hình 3. Người dùng ấn nút “Lưu dữ liệu” 4. Dữ liệu được tải xuống	

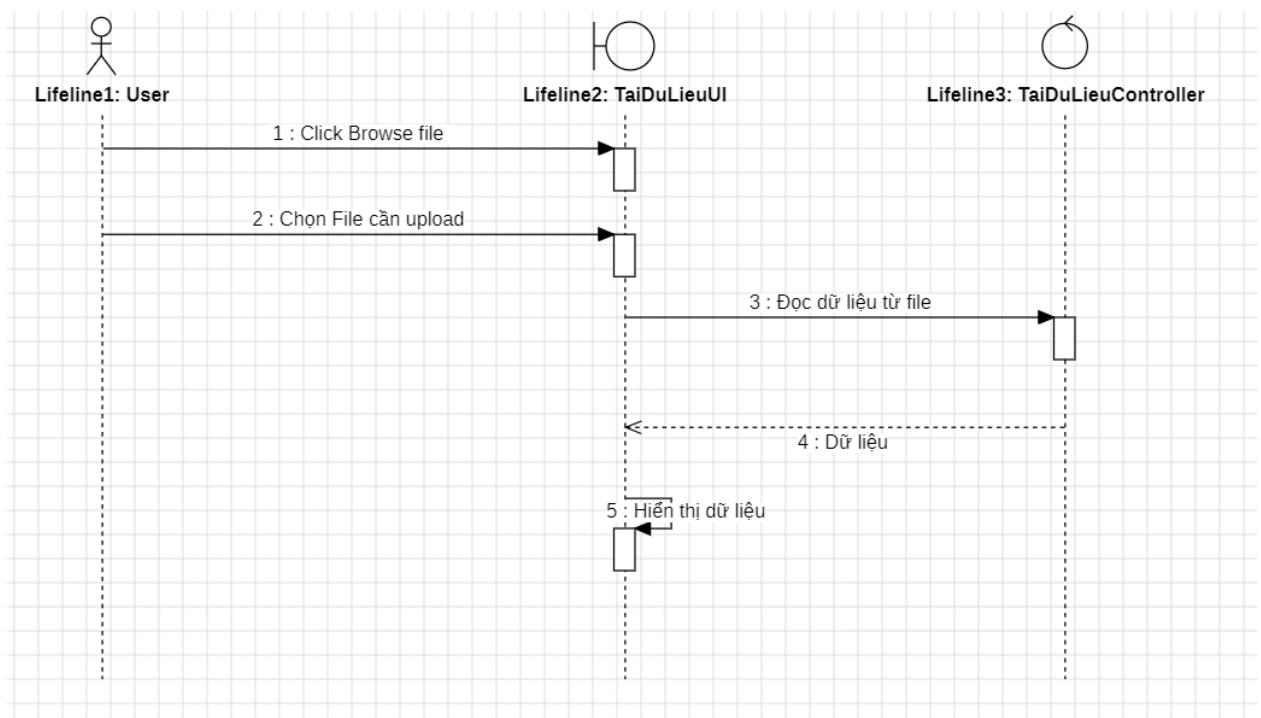
Ngoại lệ: Nếu chưa có dữ liệu dữ báo thì mục này sẽ không có gì để hiển thị.

Hậu điều kiện: Không có.

4.3.3 Phân tích các use case

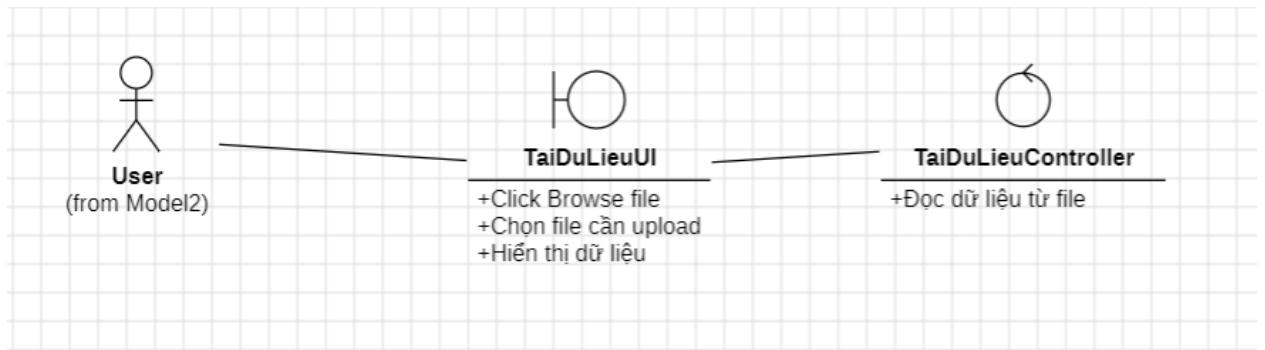
1. Use case tải dữ liệu lên

- Biểu đồ trình tự use case:



Hình 4.4: Biểu đồ trình tự use case tải dữ liệu lên.

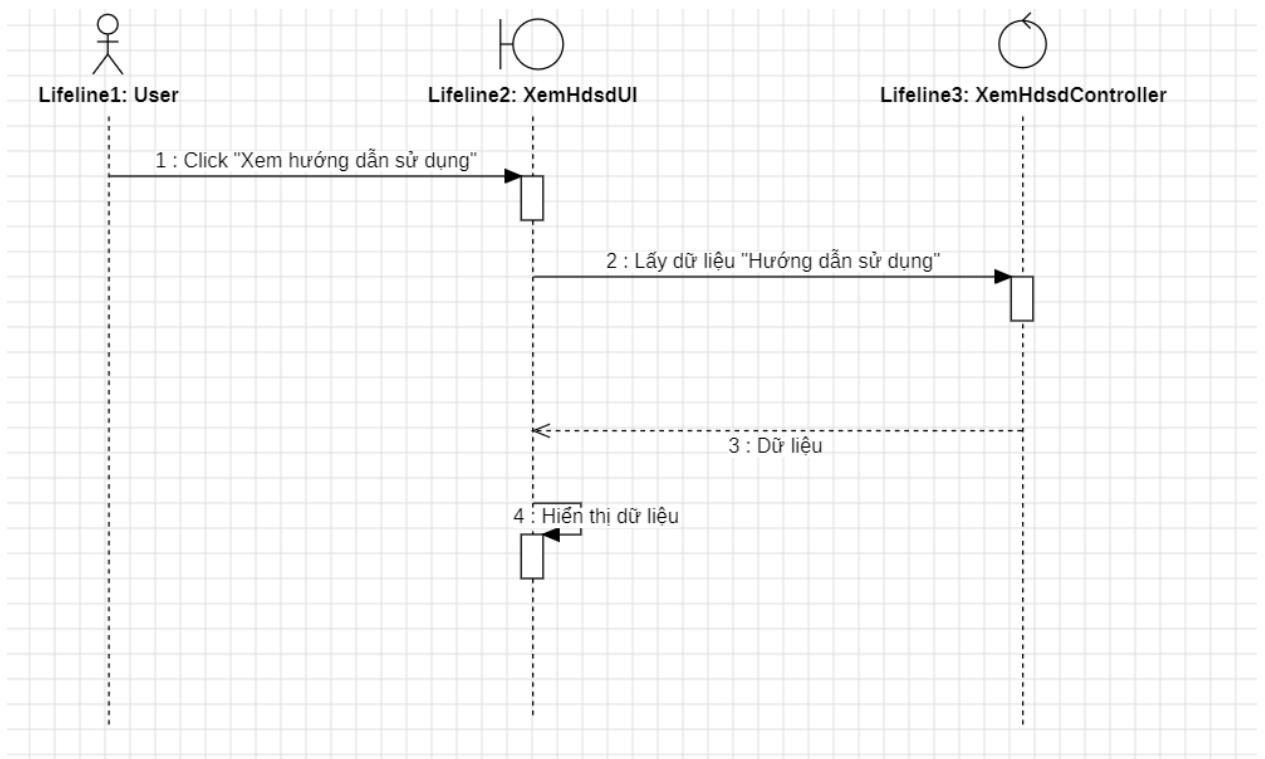
- Biểu đồ lớp phân tích:



Hình 4.5: Biểu đồ lớp phân tích use case tải dữ liệu lên.

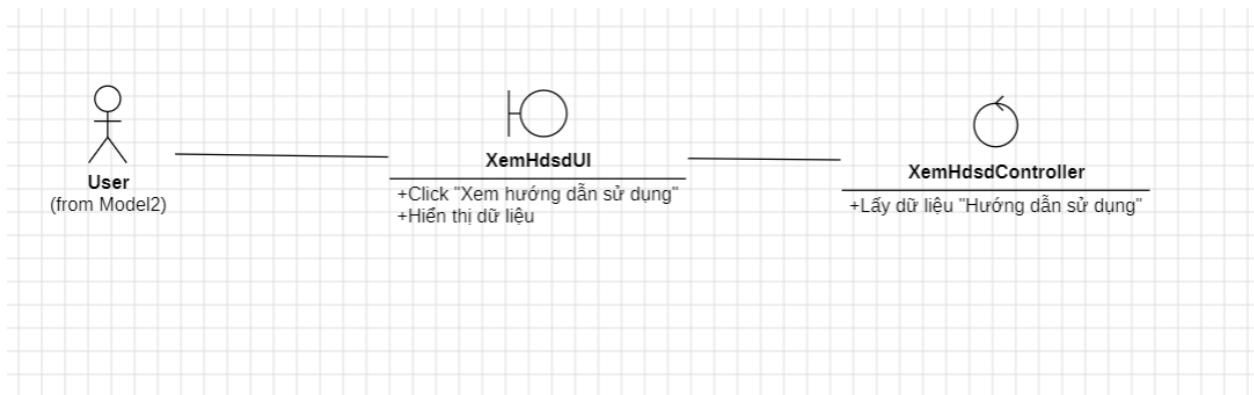
2. Use case xem hướng dẫn sử dụng phần mềm

- Biểu đồ trình tự use case:



Hình 4.6: Biểu đồ trình tự xem hướng dẫn sử dụng.

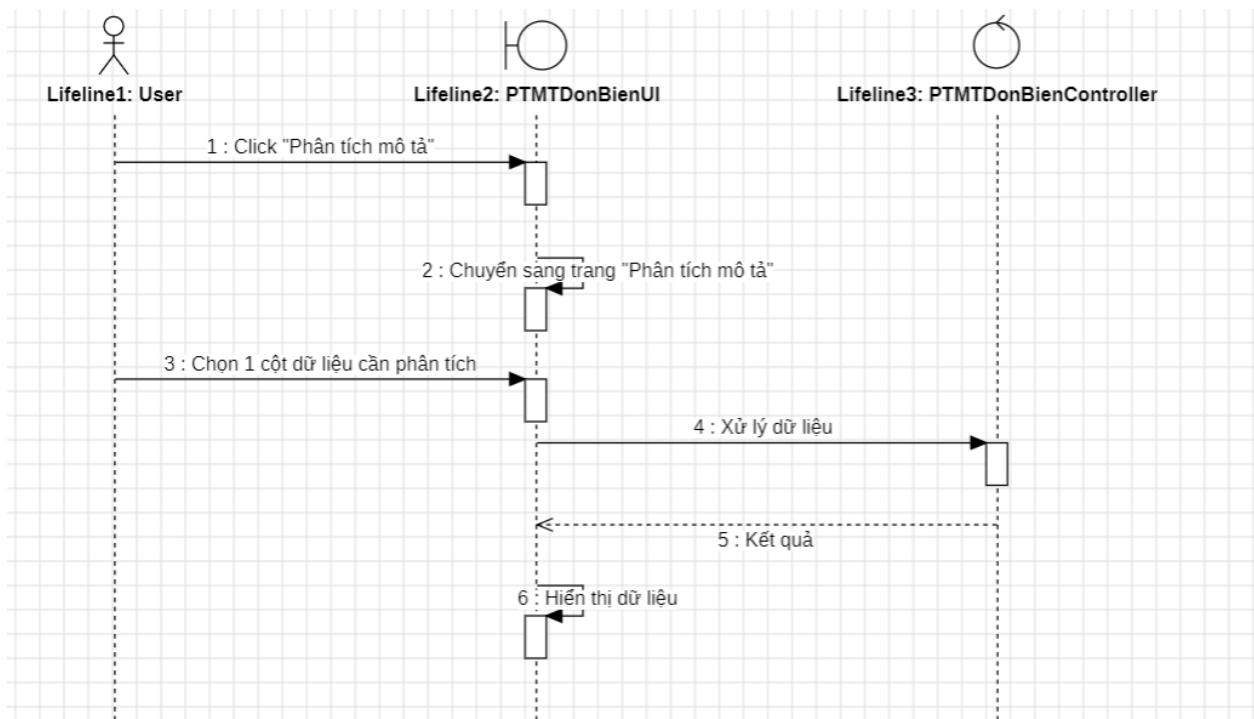
- Biểu đồ lớp phân tích:



Hình 4.7: Use case xem hướng dẫn sử dụng phần mềm.

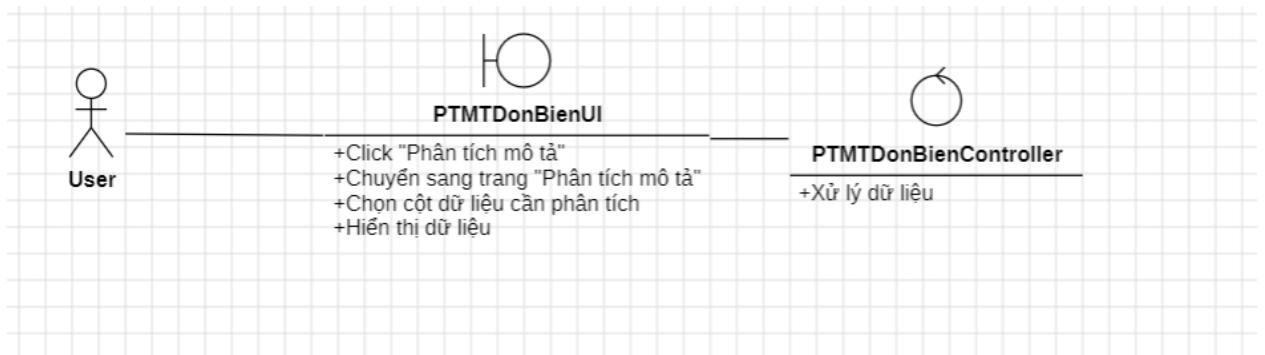
3. Use case phân tích mô tả đơn biến

- Biểu đồ trình tự use case:



Hình 4.8: Biểu đồ trình tự use case phân tích đơn biến.

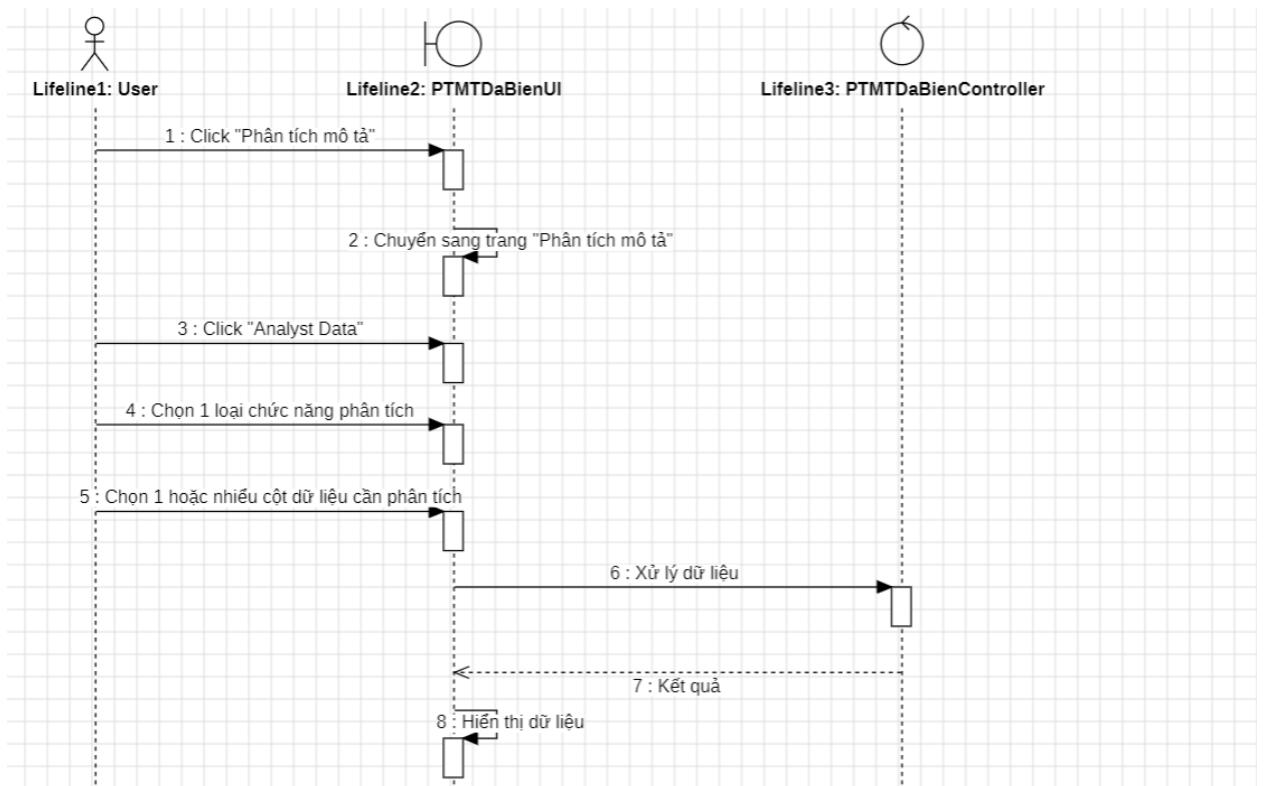
- Biểu đồ lớp phân tích:



Hình 4.9: Biểu đồ lớp phân tích use case phân tích đơn biến.

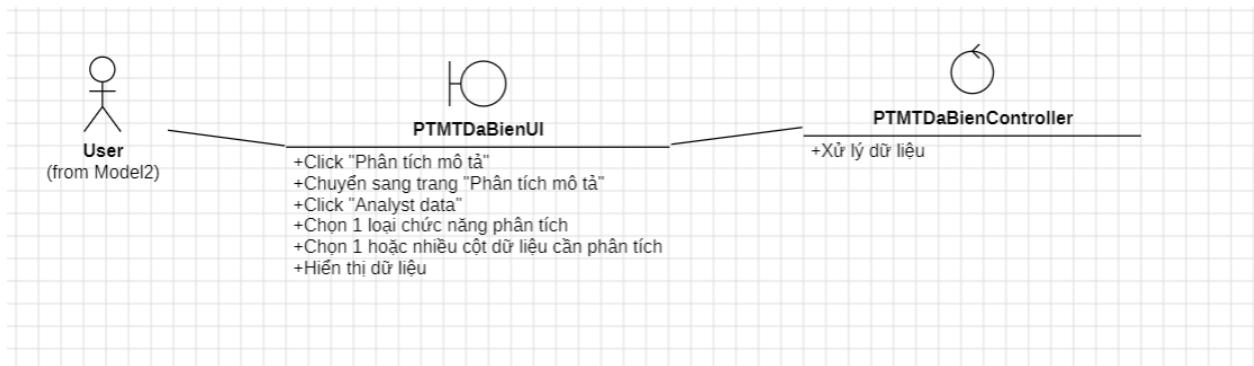
4. Use case phân tích đa biến

- Biểu đồ trình tự use case:



Hình 4.10: Biểu đồ trình tự use case phân tích đa biến.

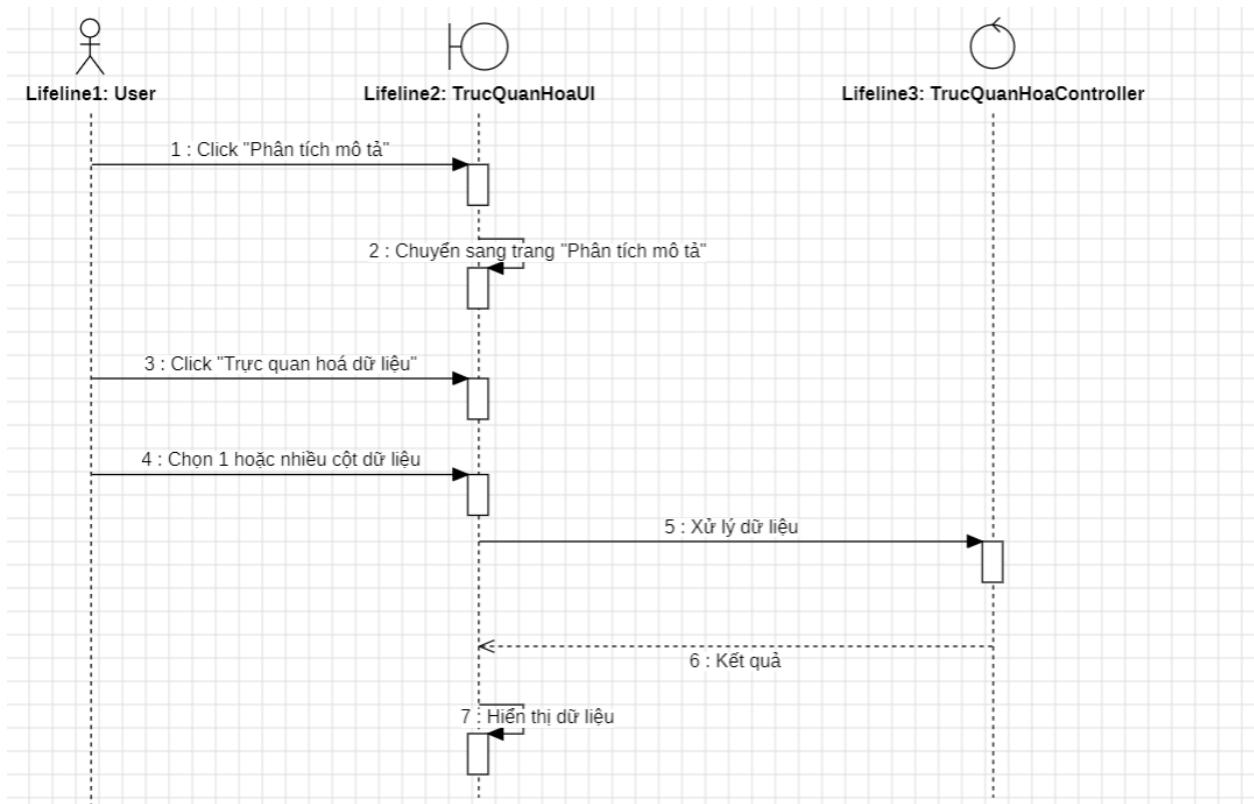
- Biểu đồ lớp phân tích:



Hình 4.11: Biểu đồ lớp phân tích use case phân tích đa biến.

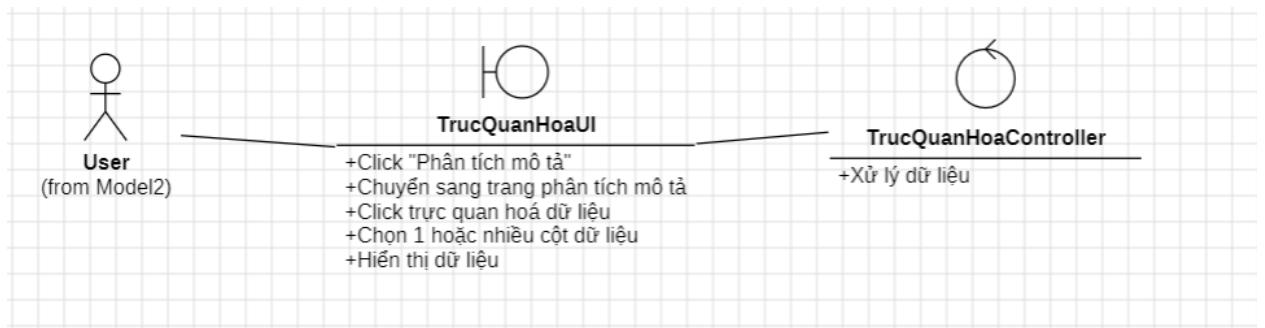
5. Use case trực quan hóa dữ liệu

- Biểu đồ trình tự use case:



Hình 4.12: Biểu đồ trình tự use case trực quan hóa dữ liệu.

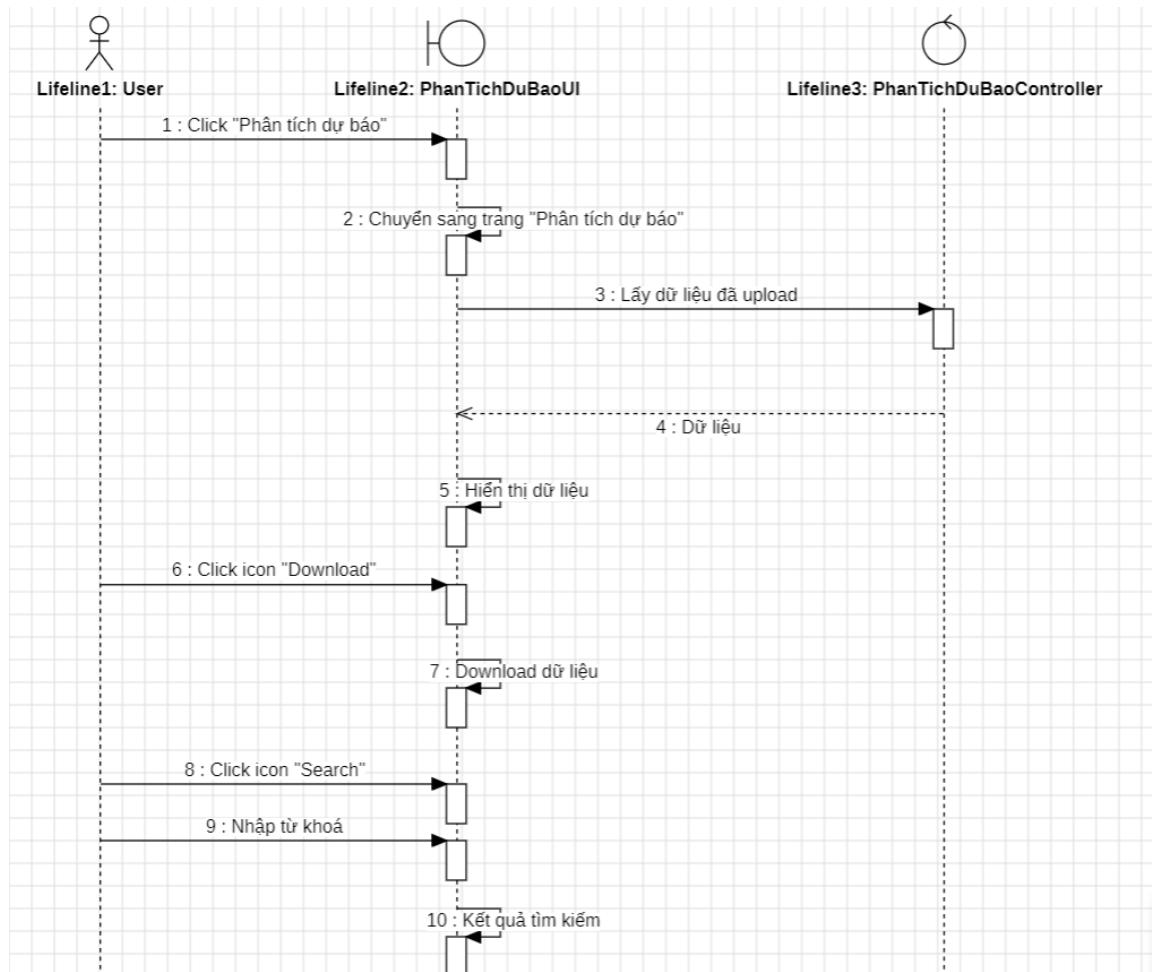
- Biểu đồ lớp phân tích:



Hình 4.13: Biểu đồ lớp phân tích use case trực quan hóa dữ liệu.

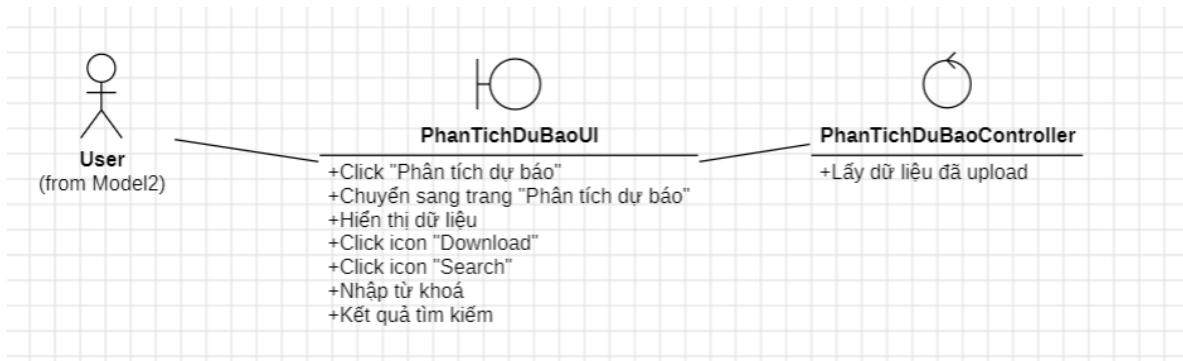
6. Use case phân tích dự báo

- Biểu đồ trình tự use case:



Hình 4.14: Biểu đồ trình tự use case phân tích dự báo.

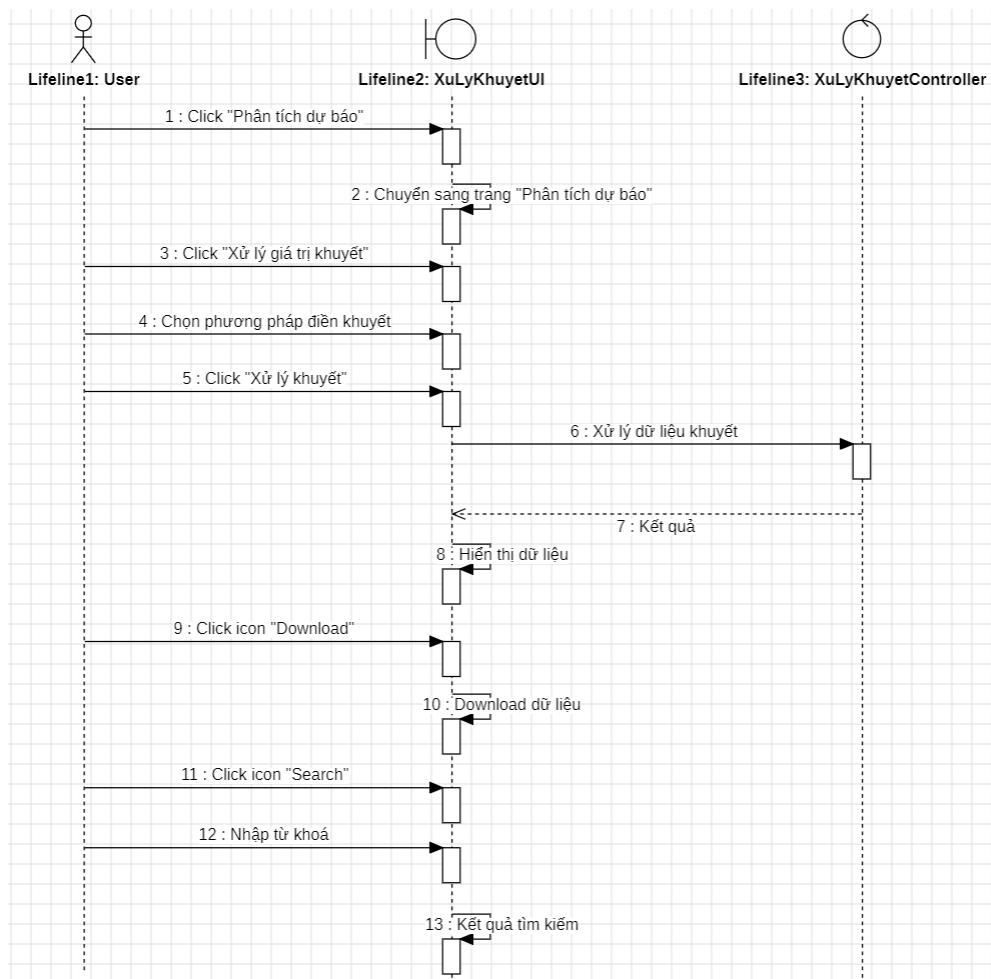
- Biểu đồ lớp phân tích:



Hình 4.15: Biểu đồ lớp phân tích use case phân tích dự báo.

7. Use case xử lý khuyết

- Biểu đồ trình tự use case:



Hình 4.16: Biểu đồ trình tự use case xử lý khuyết.

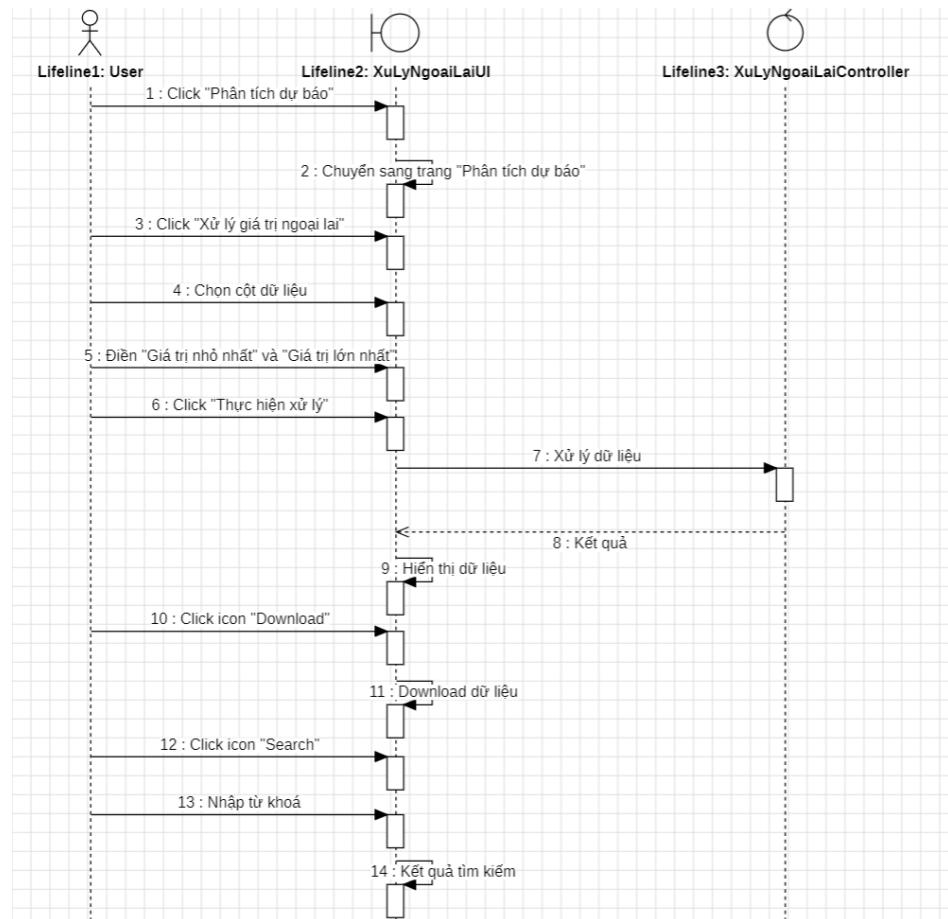
- Biểu đồ lớp phân tích:



Hình 4.17: Biểu đồ lớp phân tích use case xử lý khuyết.

8. Use case xử lý ngoại lai

- Biểu đồ trình tự use case:



Hình 4.18: Biểu đồ trình tự use case xử lý ngoại lai.

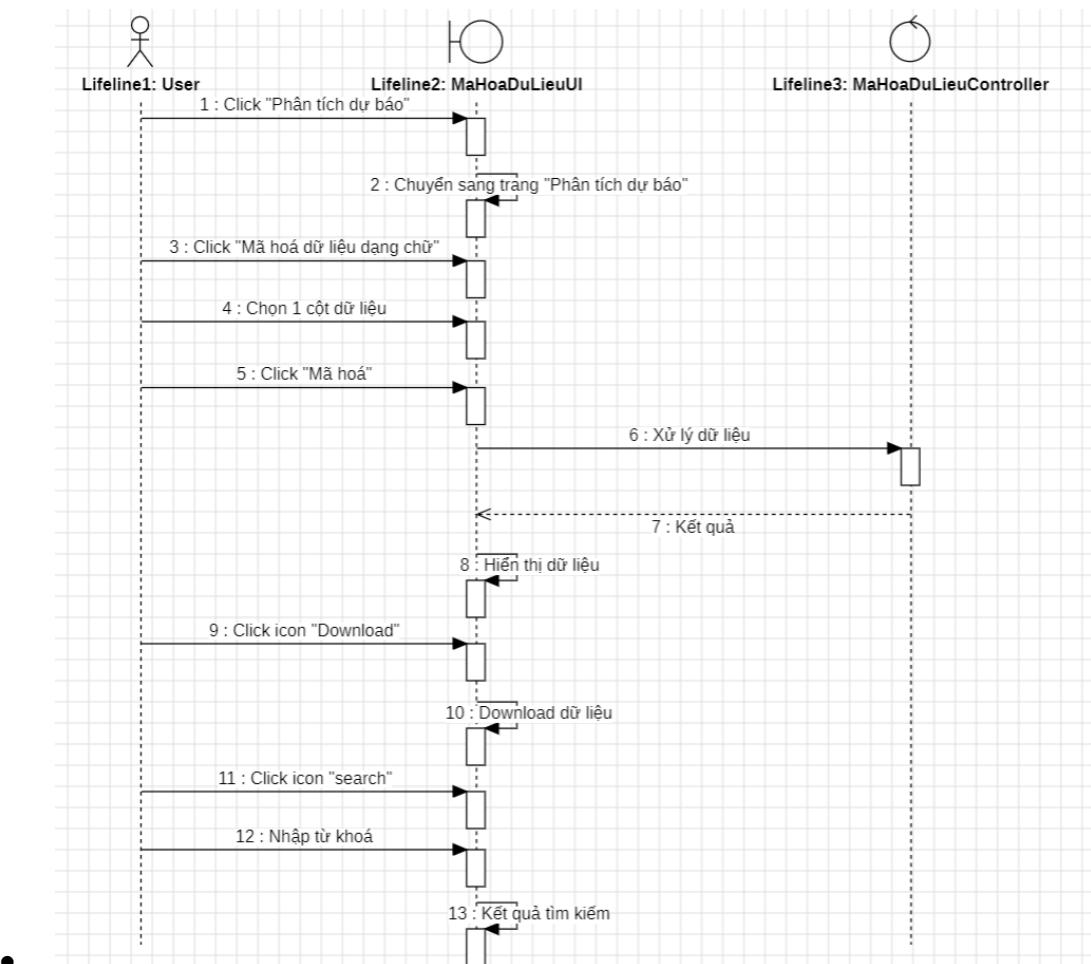
- Biểu đồ lớp phân tích:



Hình 4.19: Biểu đồ lớp phân tích use case xử lý ngoại lai.

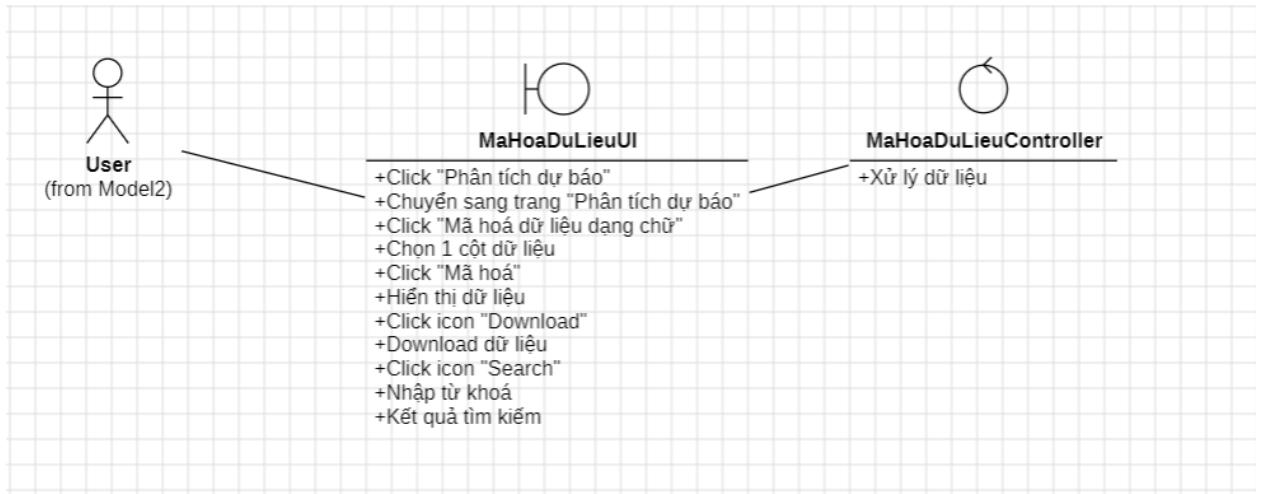
9. Use case mã hóa dữ liệu

- Biểu đồ trình tự use case:



Hình 4.20: Biểu đồ trình tự use case mã hóa dữ liệu.

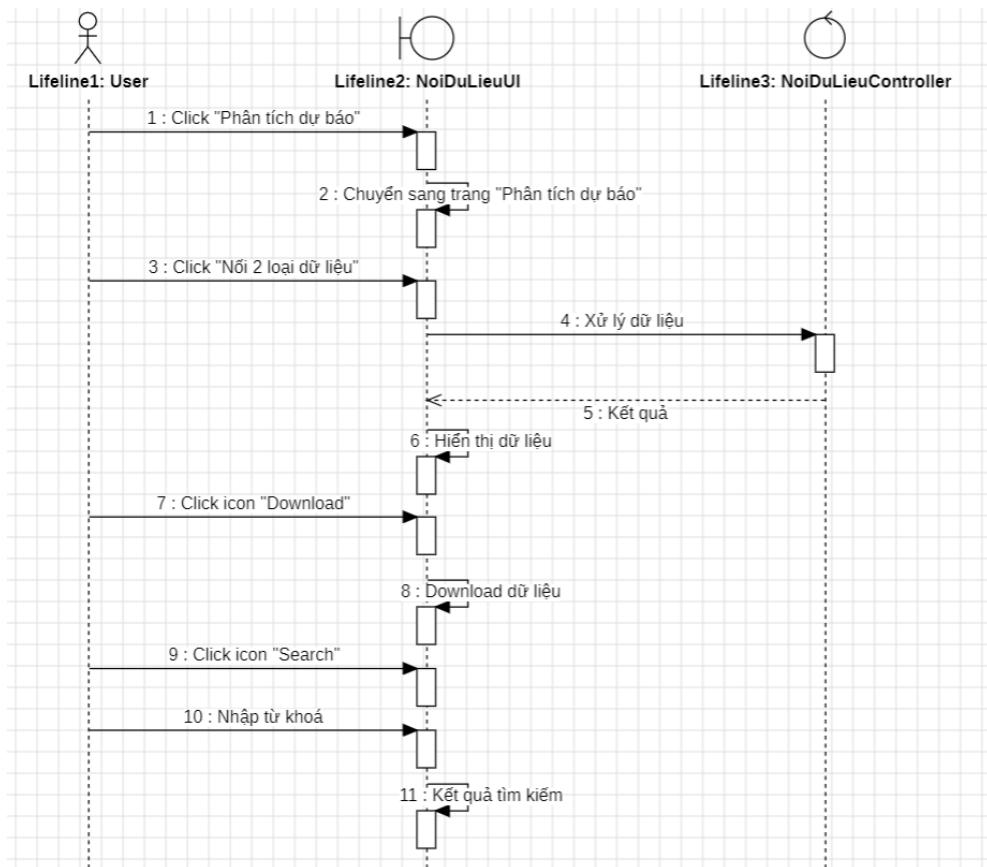
- Biểu đồ lớp phân tích:



Hình 4.21: Biểu đồ lớp phân tích use case mã hóa dữ liệu.

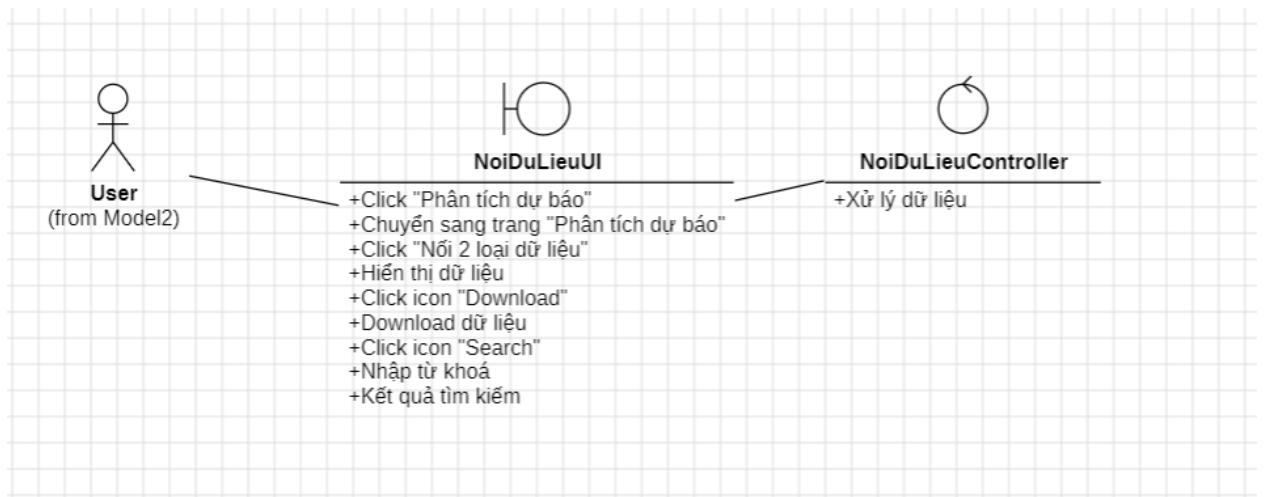
10. Use case nối dữ liệu

- Biểu đồ trình tự use case:



Hình 4.22: Biểu đồ trình tự use case nối dữ liệu.

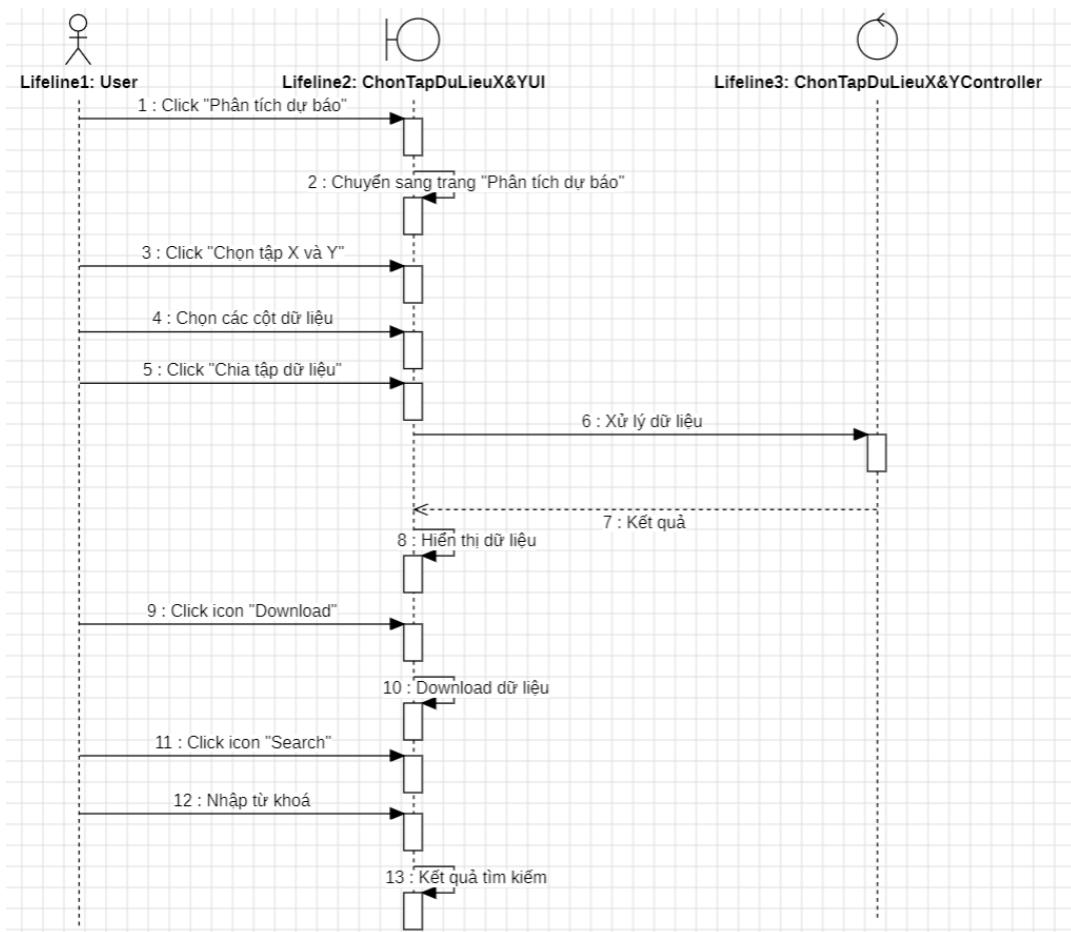
- Biểu đồ lớp phân tích:



Hình 4.23: Biểu đồ lớp phân tích use case nội dữ liệu.

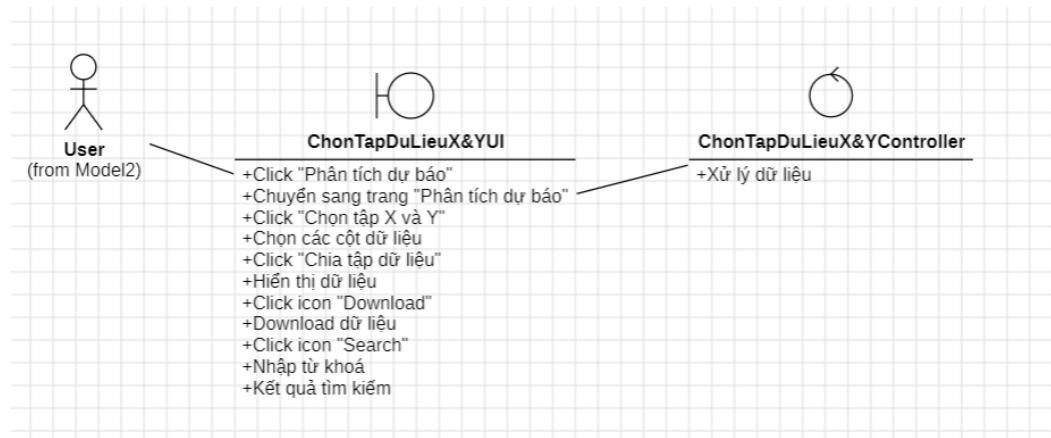
11. Use case chia tập dữ liệu

- Biểu đồ trình tự use case:



Hình 4.24: Biểu đồ trình tự use case chọn tập X-Y.

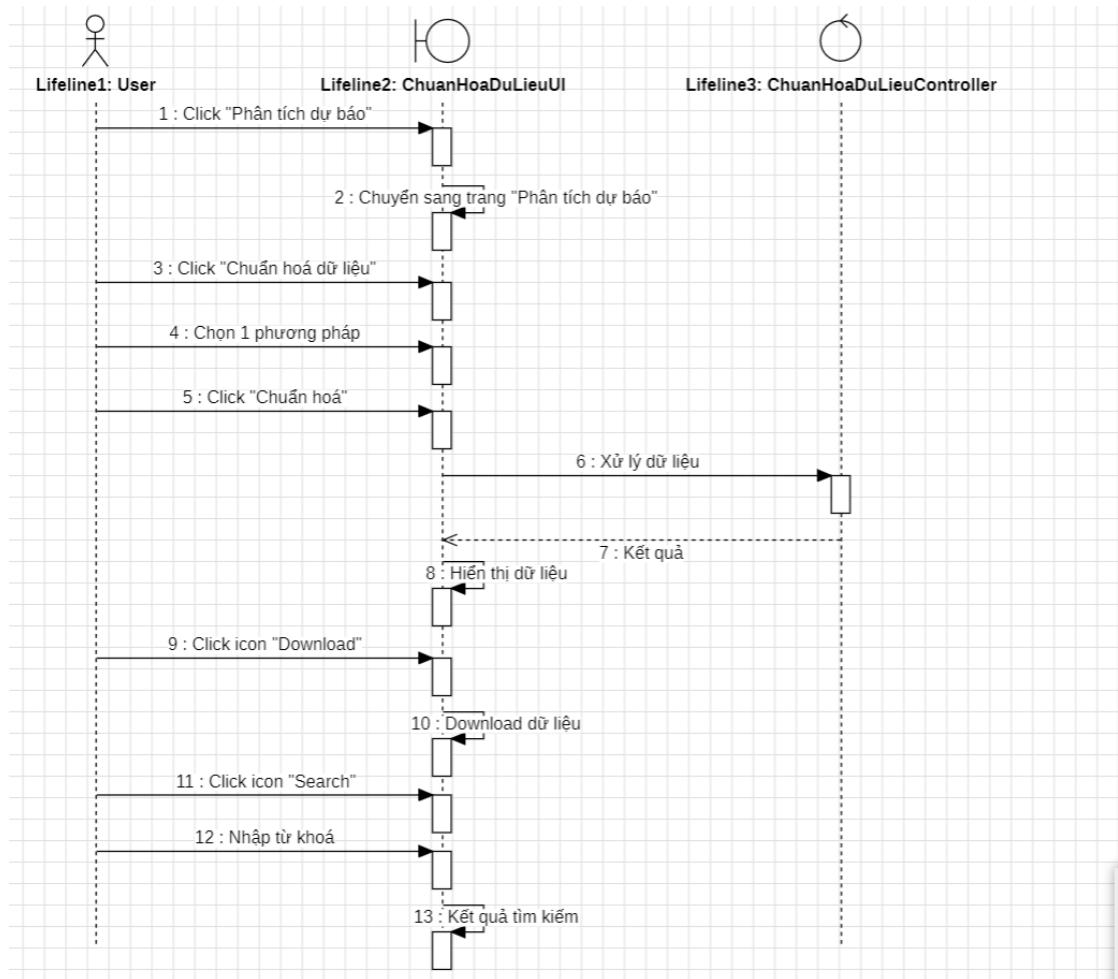
- Biểu đồ lớp phân tích:



Hình 4.25: Biểu đồ lớp phân tích use case chọn tập X-Y.

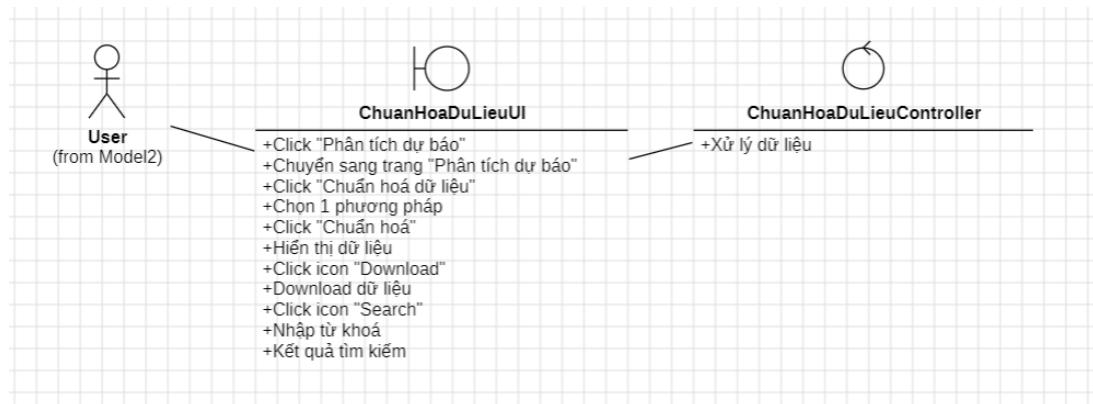
12. Use case chuẩn hóa dữ liệu

- Biểu đồ trình tự use case:



Hình 4.26: Biểu đồ trình tự use case chuẩn hóa dữ liệu.

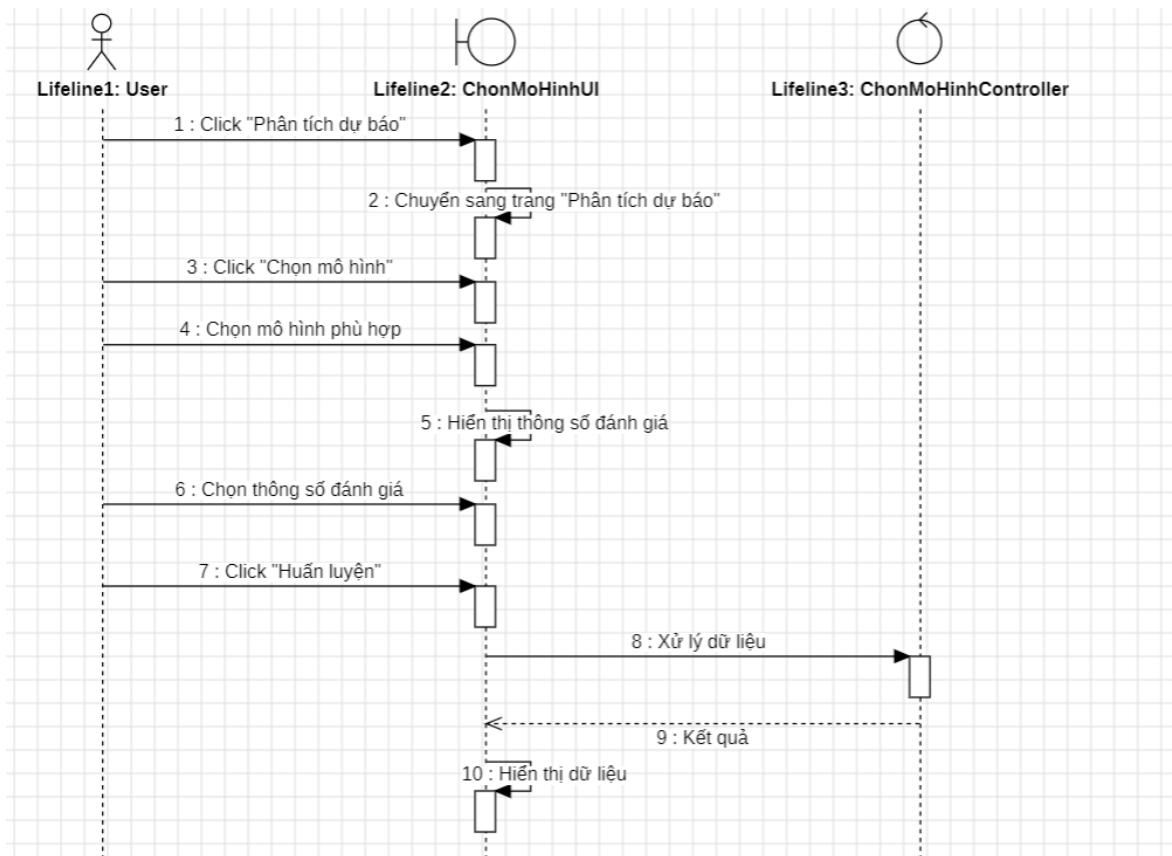
- Biểu đồ lớp phân tích:



Hình 4.27: Biểu đồ lớp phân tích use case chuẩn hóa dữ liệu.

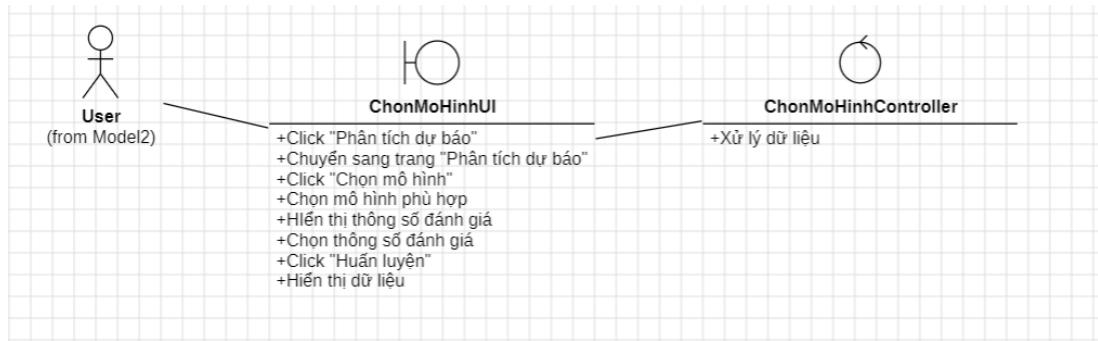
13. Use case chọn mô hình

- Biểu đồ trình tự use case:



Hình 4.28: Biểu đồ trình tự use case chọn mô hình.

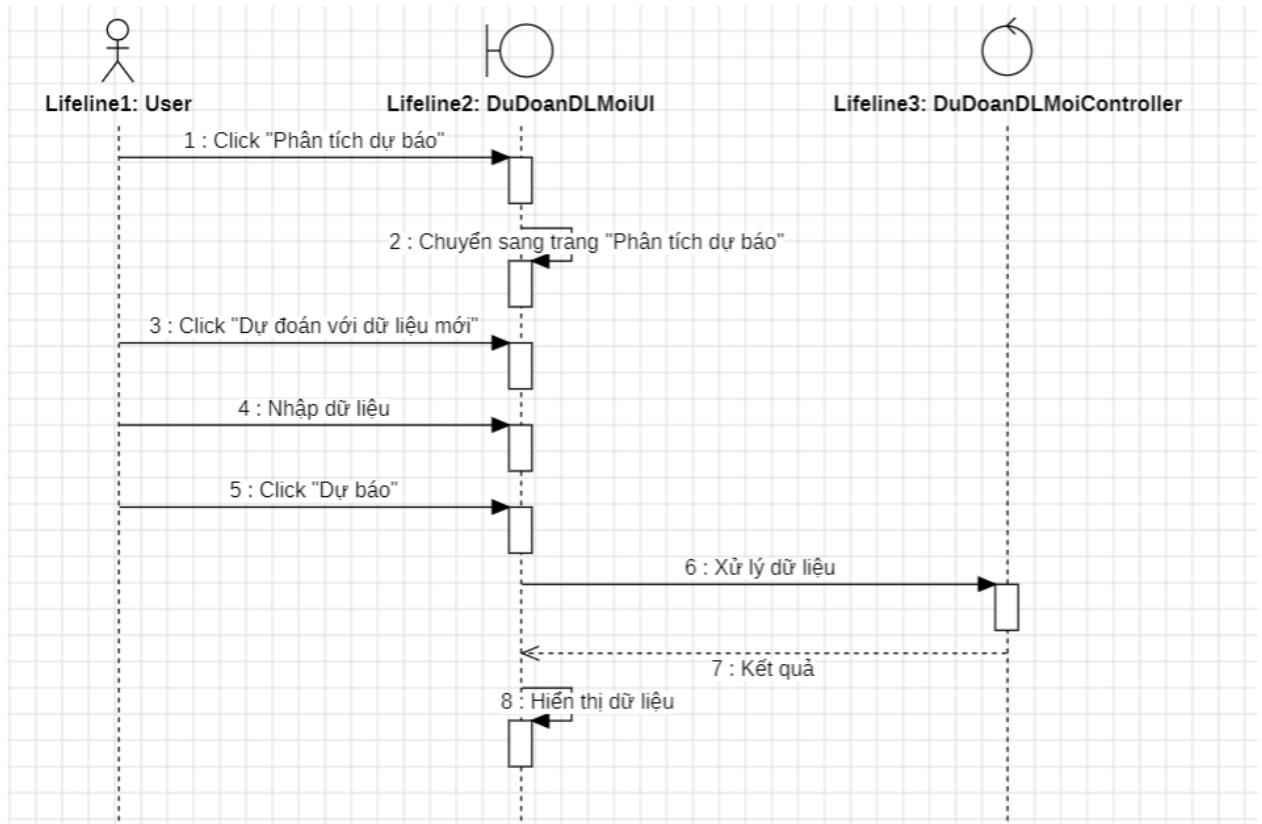
- Biểu đồ lớp phân tích:



Hình 4.29: Biểu đồ lớp phân tích use case chọn mô hình

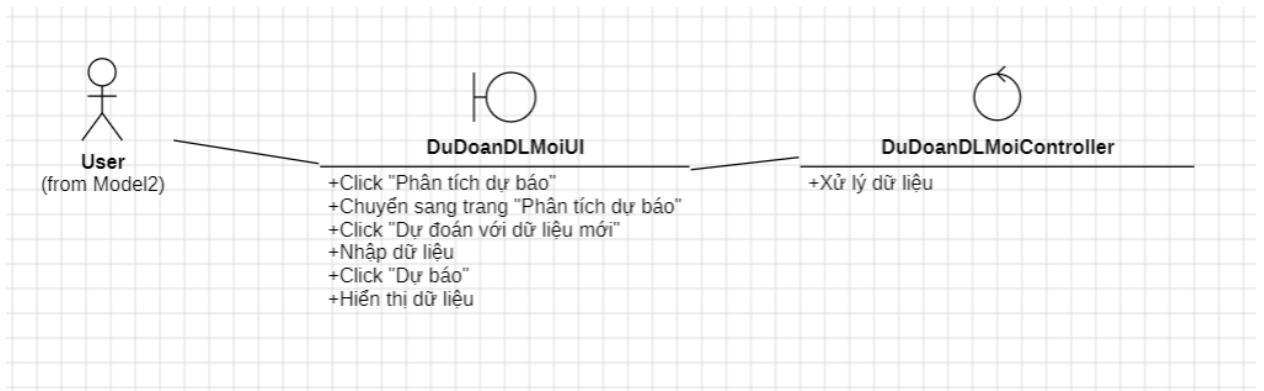
14. Use case dự đoán trên dữ liệu mới

- Biểu đồ trình tự use case:



Hình 4.30: Biểu đồ trình tự use case dự đoán dữ liệu.

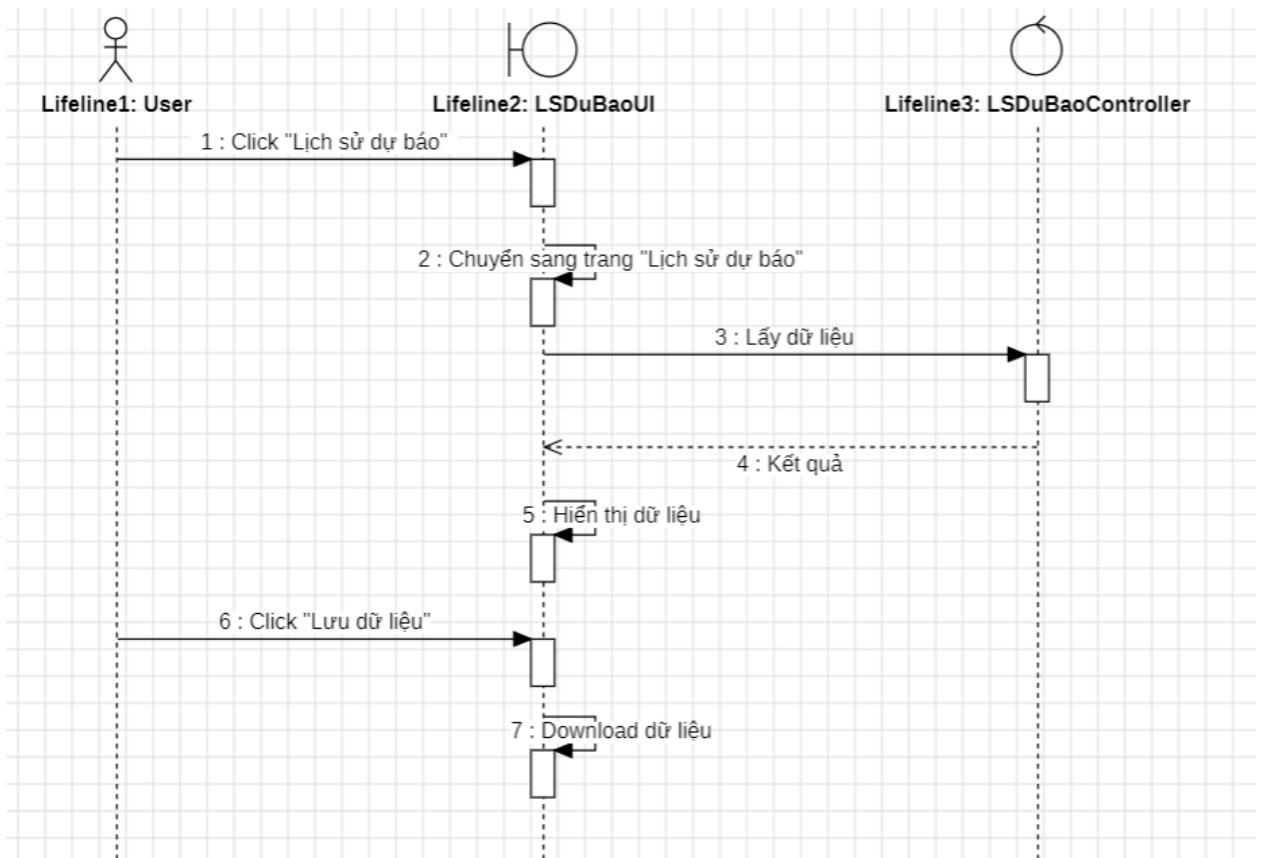
- Biểu đồ lớp phân tích:



Hình 4.31: Biểu đồ lớp phân tích use case dự đoán dữ liệu

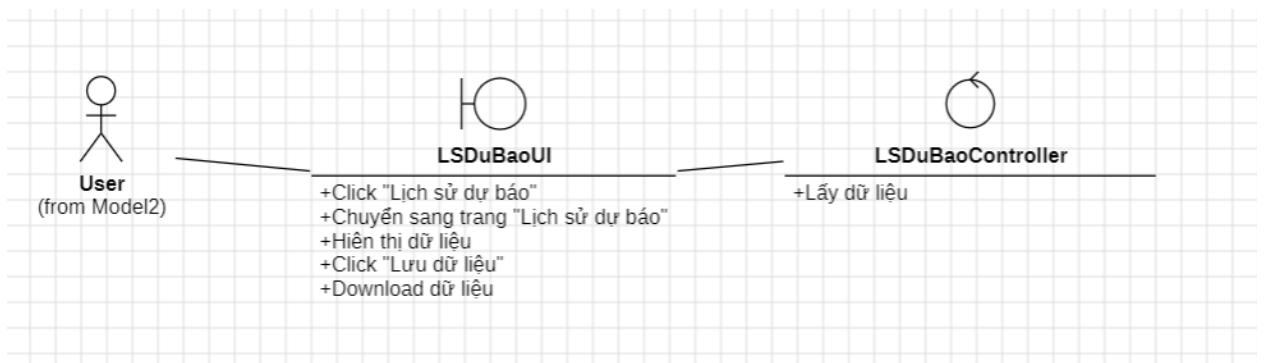
15. Use case lịch sử dự báo

- Biểu đồ trình tự use case:



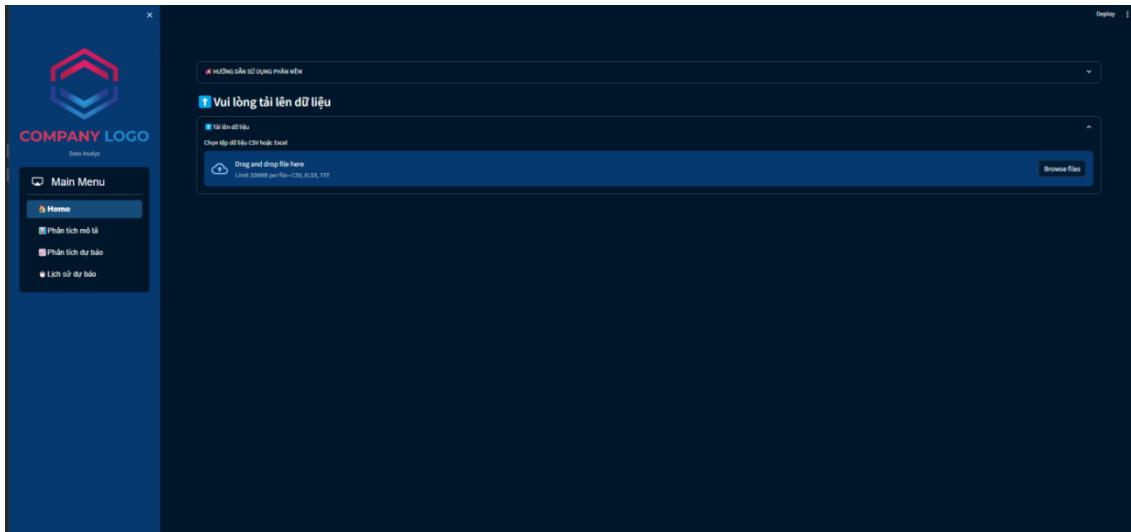
Hình 4.32: Biểu đồ trình tự use case lịch sử dự báo.

- Biểu đồ lớp phân tích:



Hình 4.33: Biểu đồ lớp phân tích use case lịch sử dự báo.

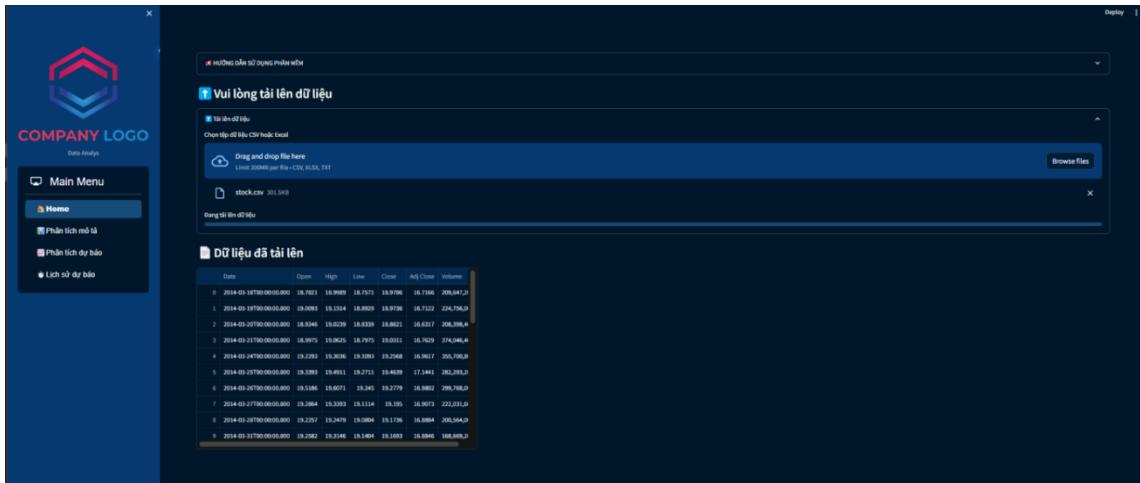
4.4 Thiết kế giao diện hệ thống



Hình 4.34: Giao diện trang chủ của ứng dụng.

Để có thể tiến hành phân tích dữ liệu, thứ quan trọng nhất chính là có được 1 tập dữ liệu để phân tích. Do đó, khi chưa upload dữ liệu lên, các giao diện được liên kết khác với trang chủ “Home” như “Phân tích mô tả”, “Phân tích dự báo”, “Lịch sử dự báo” sẽ không có dữ liệu để hiển thị.

Do vậy, khi upload dữ liệu thành công, ta mới có thể khám phá được nhiều tính năng hơn đến từ phần mềm.

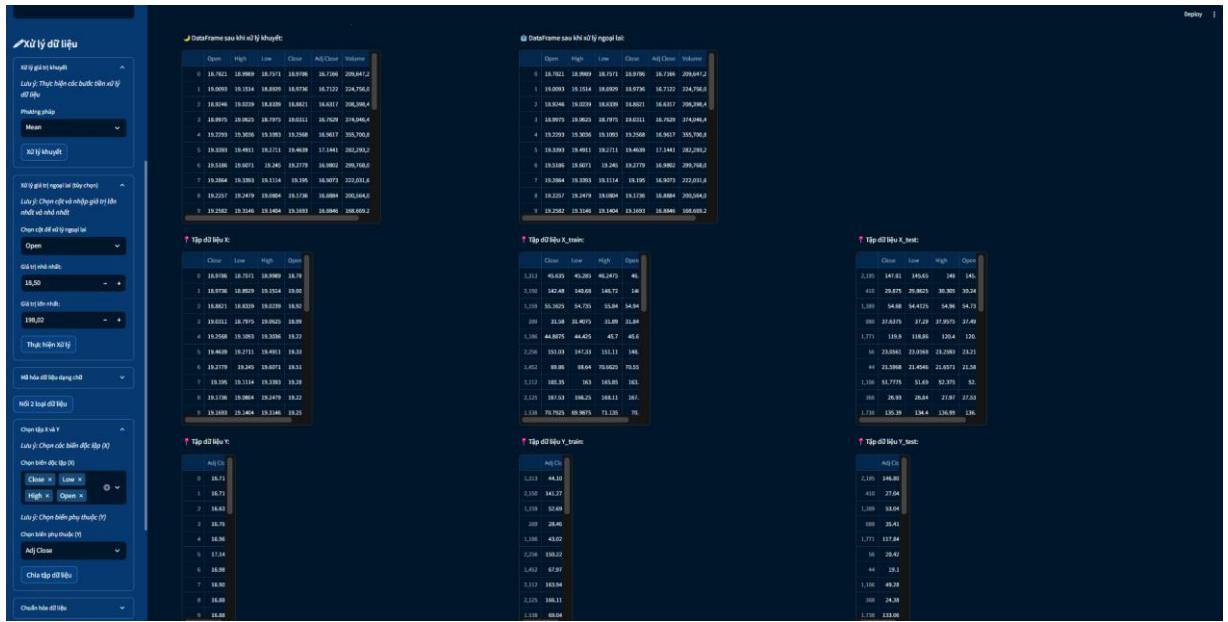


Hình 4.35: Giao diện trang chủ của ứng dụng khi đã upload dữ liệu.

Khi đã upload dữ liệu, các liên kết trang khác sẽ hiển thị theo đúng chức năng của nó



Hình 4.36: Giao diện của tính năng phân tích mô tả.



Hình 4.37: Giao diện của tính năng phân tích dự báo.

Sau các lần dự báo, ta có thể tra cứu lịch sử và lưu lịch sử dự báo dưới dạng file json. Điều quan trọng hơn, dữ liệu này có thể được coi là một API public trên phần mềm giúp cho người dùng có thể download dữ liệu họ đã dự báo một cách dễ dàng.

Lịch sử dự báo:

```
[
  {
    "0": {
      "time": "09/05/2024 13:21:07",
      "value": {
        "Dữ liệu nhập vào": [
          0: 99,
          1: 99,
          2: 99,
          3: 99
        ],
        "Dự Báo": "97.30268027499312"
      }
    },
    "1": {
      "time": "09/05/2024 13:21:25",
      "value": {
        "Dữ liệu nhập vào": [
          0: 111,
          1: 111,
          2: 111,
          3: 111
        ],
        "Dự Báo": "109.4226070105076"
      }
    },
    "2": {
      "time": "09/05/2024 13:21:31",
      "value": {
        "Dữ liệu nhập vào": [
          0: 222,
          1: 222,
          2: 222,
          3: 222
        ],
        "Dự Báo": "221.5319293140163"
      }
    }
  ]
]

Lưu dữ liệu
```

- 09/05/2024 13:21:07: {'Dữ liệu nhập vào': [99.0, 99.0, 99.0, 99.0], 'Dự Báo': '97.30268027499312'}
- 09/05/2024 13:21:25: {'Dữ liệu nhập vào': [111.0, 111.0, 111.0, 111.0], 'Dự Báo': '109.4226070105076'}
- 09/05/2024 13:21:31: {'Dữ liệu nhập vào': [222.0, 222.0, 222.0, 222.0], 'Dự Báo': '221.5319293140163'}

Hình 4.38: Dữ liệu lịch sử dự báo.

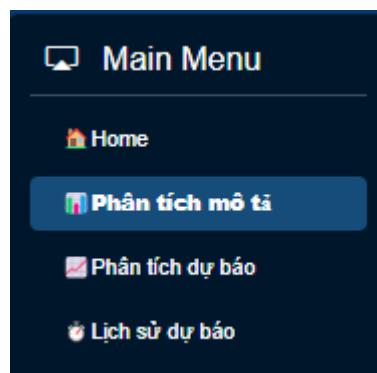
4.5 Các chức năng của hệ thống

Hệ thống có chức năng chính là phân tích dữ liệu bao gồm phân tích mô tả và phân tích dự báo dựa trên tập dữ liệu đơn giản mà người dùng upload lên.

Hệ thống phù hợp những người bắt đầu làm quen với phân tích dữ liệu, làm quen với machine learning. Hệ thống là sự kết hợp giữa phân tích dữ liệu, machine learning và trực quan hóa dữ liệu.

Với sự phát triển của AI cũng như ngày càng có một lượng lớn data được lưu hành, việc phân tích dữ liệu và chuyển đổi dữ liệu (từ tập dữ liệu này để có thể có được một tập dữ liệu khác có giá trị hơn) đang ngày càng quan trọng và phát huy được nhiều vai trò hơn. Do vậy, hệ thống cũng cho phép người dùng lưu lại những tập dữ liệu sau khi chuyển đổi, chỉnh sửa, kết quả của tập dữ liệu sau khi áp dụng học máy để người dùng có thể thu được những kết quả thực sự có giá trị.

Đầu tiên, trang web sẽ chia làm 4 mục bao gồm “Home”, “Phân tích mô tả”, “Phân tích dự báo”, “Lịch sử dự báo”. Mỗi mục khi được click chuột sẽ chuyển hướng đến một giao diện khác nhau, phù hợp với chức năng của nó.



Hình 4.39: Chức năng upload dữ liệu

```

with st.sidebar:
    st.sidebar.image("../img/image.png", caption='Data Analys')
    selected = option_menu(
        menu_title="Main Menu",
        options=[['🏠 Home', '📊 Phân tích mô tả', '📈 Phân tích dự báo', '📅 Lịch sử dự báo'],
        icons=['🏠', '📊', '📈', '📅'],
        menu_icon="cast",
        default_index=0
    )

try:
    if selected == "🏠 Home":
        st.session_state.data = upload_data()
    if selected == "📊 Phân tích mô tả" and 'data' in st.session_state:
        st.subheader("💡 Bắt đầu quá trình phân tích mô tả")
        show_statistics(st.session_state.data)
        visualize_data(st.session_state.data)
    if selected == "📈 Phân tích dự báo" and 'data' in st.session_state:
        st.subheader("💡 Bắt đầu quá trình phân tích dự báo")
        preprocessing(st.session_state.data)
    if selected == "📅 Lịch sử dự báo":
        history()
except:
    pass

if 'data' not in st.session_state:
    st.subheader("Chưa có dữ liệu để phân tích!")

```

Hình 4.40: Source code optine menu

Chức năng upload dữ liệu

Nói về chức năng upload dữ liệu, chức năng này nằm ở mục ‘Home’, cho phép người dùng upload dữ liệu gốc muốn chỉnh sửa, phân tích lên trên web.



The screenshot shows a file upload interface. At the top, there's a message: 'Chọn tệp dữ liệu CSV hoặc Excel' and 'Drag and drop file here. Limit 20MB per file - CSV, XLSX, TXT'. Below this is a preview section titled 'Dữ liệu đã tải lên' showing a table with 10 rows of financial data:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2014-03-18T00:00:00.000	18.782	18.999	18.791	18.976	18.736	209,647,2
1	2014-03-19T00:00:00.000	19.095	19.154	18.809	18.976	18.732	204,796,5
2	2014-03-20T00:00:00.000	18.924	19.079	18.859	18.821	18.837	208,398,4
3	2014-03-21T00:00:00.000	18.997	19.062	18.797	19.011	18.762	214,046,4
4	2014-03-24T00:00:00.000	18.229	19.303	18.193	19.268	18.961	355,700,8
5	2014-03-25T00:00:00.000	19.589	19.491	19.271	19.489	17.144	282,293,2
6	2014-03-26T00:00:00.000	19.518	19.607	19.445	19.779	18.999	299,168,0
7	2014-03-27T00:00:00.000	18.264	19.339	18.114	19.195	18.907	222,031,6
8	2014-03-28T00:00:00.000	18.257	19.2479	18.084	19.176	18.8884	203,564,0
9	2014-03-29T00:00:00.000	19.2982	19.3146	18.1484	19.3193	18.8866	168,693,2

Hình 4.41: Chức năng upload dữ liệu.

```

19  def upload_data():
20      st.subheader('Vui lòng tải lên dữ liệu')
21
22  with st.expander("Tải lên dữ liệu"):
23      uploaded_file = st.file_uploader("Chọn tệp dữ liệu CSV hoặc Excel", type=['csv', 'xlsx', 'txt'])
24  if uploaded_file is not None:
25      try:
26          # Hiển thị thanh tiến trình
27          progress_bar = st.progress(0)
28
29          # Giả lập việc tải lên trong 1 giây
30          for percent_complete in range(100):
31              time.sleep(0.000) # Giảm thời gian sleep để tăng tốc độ hiển thị thanh tiến trình
32              progress_bar.progress(percent_complete + 1, text='Đang tải lên dữ liệu')
33
34          if uploaded_file.name.endswith('.csv'):
35              st.session_state.df = pd.read_csv(uploaded_file)
36          elif uploaded_file.name.endswith('.xlsx'):
37              st.session_state.df = pd.read_excel(uploaded_file)
38          else:
39              st.session_state.df = pd.read_csv(uploaded_file, delimiter='\t')
40
41      except Exception as e:
42          st.error(f"Đã xảy ra lỗi khi đọc tệp: {e}")
43  if "df" in st.session_state:
44      st.subheader('Dữ liệu đã tải lên')
45      st.write(st.session_state.df)
46  return st.session_state.df

```

Hình 4.42: Source code chức năng upload dữ liệu.

Sau khi upload dữ liệu thành công, ta chuyển đến mục “Phân tích mô tả”.

Trong mục này, chức năng chính là để phân tích mô tả tập dữ liệu đã upload bao gồm phân tích thống kê và trực quan hóa dữ liệu.

Chức năng phân tích mô tả

Trong phân tích mô tả, ta sẽ có 3 chức năng chính là: Phân tích thống kê đơn biến, phân tích thống kê đa biến, trực quan hóa dữ liệu. Phân tích thống kê đơn biến, ta tiến hành phân tích trên 1 cột dữ liệu đã chọn và hiển thị các thông số thống kê như min, max, sum, mean, count null lên màn hình. Để có thể thực hiện chức năng này, người dùng cần chọn cột dữ liệu để tiến hành phân tích. Nếu không có sự lựa chọn nào, hệ thống sẽ tự động chọn cột đầu tiên trong tập dữ liệu.

Bắt đầu quá trình phân tích mô tả	LongestShell	Min of LongestShell	Max of LongestShell	Sum of LongestShell	Mean of LongestShell	Count null of LongestShell
Cột được chọn LongestShell	0	1	2,189	1	0	

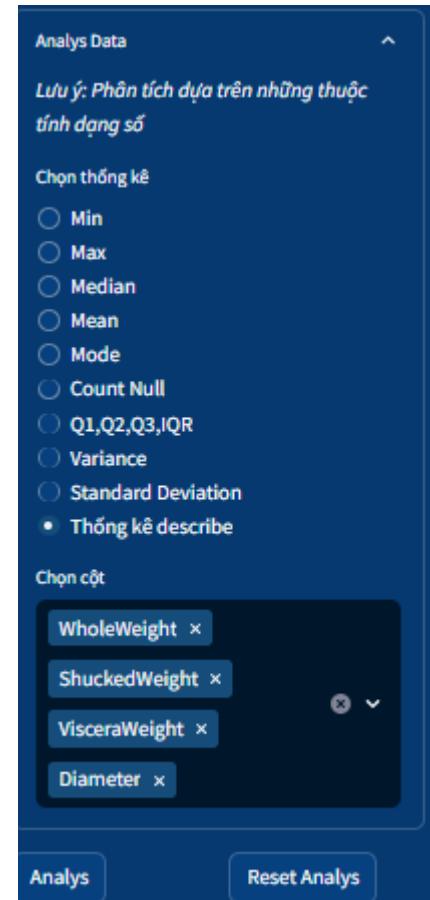
Hình 4.43: Chức năng phân tích mô tả đơn biến.

```
def show_statistics(df):
    st.session_state.numerical_columns = df.select_dtypes(include=['number']).columns
    ## decor màn chính
    total1,total2,total3,total4,total5,total6 = st.columns(6, gap='small')
    with total1:
        st.session_state.select_column = st.selectbox('Chọn 1 cột',label_visibility='collapsed', options=st.session_state.numerical_columns, key='select_columns')
        if not st.session_state.select_column:
            st.session_state.select_column = "None"
            st.warning("Vui lòng chọn một cột.")
        st.metric(label= "Cột được chọn", value=f"{st.session_state.select_column}")

    if st.session_state.select_column != "None":
        st.session_state.min = (df[st.session_state.select_column]).min()
        st.session_state.max = (df[st.session_state.select_column]).max()
        st.session_state.sum = (df[st.session_state.select_column]).sum()
        st.session_state.mean = (df[st.session_state.select_column]).mean()
        st.session_state.count_null = (df[st.session_state.select_column]).isnull().sum()
    with total2:
        st.info(f"Min of {st.session_state.select_column}",icon='star')
        st.metric(label= f"Min of {st.session_state.select_column}", value=f"{st.session_state.min:.0f}")
    with total3:
        st.info(f"Max of {st.session_state.select_column}",icon='star')
        st.metric(label= f"Max of {st.session_state.select_column}", value=f"{st.session_state.max:.0f}")
    with total4:
        st.info(f"Sum of {st.session_state.select_column}",icon='star')
        st.metric(label= f"Sum of {st.session_state.select_column}", value=f"{st.session_state.sum:.0f}")
    with total5:
        st.info(f"Mean of {st.session_state.select_column}",icon='star')
        st.metric(label= f"Mean of {st.session_state.select_column}", value=f"{st.session_state.mean:.0f}")
    with total6:
        st.info(f"Count null of {st.session_state.select_column}",icon='star')
        st.metric(label= f"Count null of {st.session_state.select_column}", value=f"{st.session_state.count_null:.0f}")
    else:
        with total2:
            st.info(f"Min of {st.session_state.select_column}",icon='star')
            st.metric(label= f"Min of {st.session_state.select_column}", value=f"{0}")
        with total3:
            st.info(f"Max of {st.session_state.select_column}",icon='star')
            st.metric(label= f"Max of {st.session_state.select_column}", value=f"{0}")
        with total4:
            st.info(f"Sum of {st.session_state.select_column}",icon='star')
            st.metric(label= f"Sum of {st.session_state.select_column}", value=f"{0}")
        with total5:
            st.info(f"Mean of {st.session_state.select_column}",icon='star')
            st.metric(label= f"Mean of {st.session_state.select_column}", value=f"{0}")
        with total6:
            st.info(f"Count null of {st.session_state.select_column}",icon='star')
            st.metric(label= f"Count null of {st.session_state.select_column}", value=f"{0}")
    style_metric_cards(border_color="blue",background_color="#00172B",border_left_color="blue")
    st.markdown("-----")
```

Hình 4.44: Source code chức năng phân tích mô tả đơn biến.

Phân tích thống kê đa biến, ta tiến hành phân tích trên 1 hoặc nhiều cột dữ liệu đã chọn và hiển thị các thông số thống kê như min, max, median, mean, mode, count null, Q1, Q2, Q3, IQR, variance, stdev, thống kê describe lên màn hình. Để có thể thực hiện chức năng này, người dùng cần chọn cột dữ liệu để tiến hành phân tích. Nếu không có sự lựa chọn nào về thông số thống kê, hệ thống sẽ tự động để là min. Nếu chưa có cột dữ liệu nào được lựa chọn, hệ thống sẽ không hiển thị nút “Analys”. Dữ liệu thống kê chỉ hiển thị khi nút “Analys” được kích hoạt.



Thống kê dữ liệu				
	WholeWeight	ShuckedWeight	VisceraWeight	Diameter
count	4,177	4,177	4,177	4,177
mean	0.8287	0.3594	0.1806	0.4079
std	0.4904	0.222	0.1096	0.0992
min	0.002	0.001	0.0005	0.055
25%	0.4415	0.186	0.0935	0.35
50%	0.7995	0.336	0.171	0.425
75%	1.153	0.502	0.253	0.48
max	2.8255	1.488	0.76	0.65

Hình 4.45: Chức năng phân tích mô tả đa biến.

```

with st.sidebar:
    st.title('Phân tích mô tả')
    # Kiểm tra xem DataFrame có dữ liệu dạng số không

if not st.session_state.numerical_columns.empty:
    # Tao expander mới trong sidebar
    with st.sidebar.expander("Analys Data"):
        st.markdown("Lưu ý: Phân tích dựa trên những thuộc tính dạng số")
        # Thêm radio button cho các thông kê muốn hiển thị
        selected_statistic = st.radio("Chọn thông kê", options=['Min', 'Max', 'Mean', 'Mode', 'Count Null', 'Q1,Q2,Q3,IQR', 'Variance', 'Standard Deviation', 'Thông kê describe'])
        selected_columns = st.multiselect("Chọn cột", options=st.session_state.numerical_columns, key='select_analys')
        if not selected_columns:
            st.warning("Vui lòng chọn ít nhất một cột.")
            return

        # Xử lý hiển thị dữ liệu theo radio button được chọn
        if df is not None:
            col1, col2 = st.sidebar.columns(2)
            with col1:
                if st.button("Analys"):
                    if selected_statistic == 'Thông kê describe':
                        st.session_state.statistics_result = df[selected_columns].describe(include='all')
                    elif selected_statistic == 'Min':
                        st.session_state.statistics_result = df[selected_columns].min()
                    elif selected_statistic == 'Max':
                        st.session_state.statistics_result = df[selected_columns].max()
                    elif selected_statistic == 'Mean':
                        st.session_state.statistics_result = df[selected_columns].mean()
                    elif selected_statistic == 'Median':
                        st.session_state.statistics_result = df[selected_columns].median()
                    elif selected_statistic == 'Mode':
                        st.session_state.statistics_result = df[selected_columns].mode().iloc[0]
                    elif selected_statistic == 'Count Null':
                        st.session_state.statistics_result = df[selected_columns].isnull().sum()
                    elif selected_statistic == 'Q1,Q2,Q3,IQR':
                        quantiles = df[selected_columns].quantile([0.25, 0.5, 0.75])
                        quantiles.loc['IQR'] = quantiles.loc[0.75] - quantiles.loc[0.25]
                        st.session_state.statistics_result = quantiles
                    elif selected_statistic == 'Variance':
                        st.session_state.statistics_result = df[selected_columns].var()
                    elif selected_statistic == 'Standard Deviation':
                        st.session_state.statistics_result = df[selected_columns].std()
                    else:
                        st.session_state.statistics_result = getattr(df[selected_columns], selected_statistic.lower())()

            with col2:
                if st.button("Reset Analys"):
                    st.session_state.statistics_result = None

    # Hiển thị kết quả phân tích trên giao diện chính
    if "statistics_result" in st.session_state:
        st.write("Thông kê dữ liệu")
        st.write(st.session_state.statistics_result)
    else:
        st.warning("Không có cột nào chứa dữ liệu dạng số trong dữ liệu.")


```

Hình 4.46: Source code chức năng phân tích mô tả đa biến.

Trực quan hóa dữ liệu là chức năng người dùng có thể xem được 6 loại biểu đồ dữ liệu bao gồm biểu đồ đường, biểu đồ hộp, biểu đồ scatter, biểu đồ cột, biểu đồ heatmap, biểu đồ pie. Dựa trên 1 cột dữ liệu được hệ thống chọn random, người dùng có thể thấy được 6 biểu đồ trên khi bắt đầu truy cập vào giao diện của mục “Phân tích mô tả”. Tuy nhiên, để có thể xem được biểu đồ của một hoặc nhiều cột trong tập dữ liệu đó, ta có thể chọn tên các cột ở mục “Trực quan hóa dữ liệu”.



Hình 4.47: *Chức năng trực quan hóa dữ liệu.*

```

def visualize_data(df):
    st.session_state.numerical_columns = df.select_dtypes(include=['number']).columns
    st.session_state.df_number = df[st.session_state.numerical_columns]
    if not st.session_state.numerical_columns.empty:
        with st.sidebar.expander("Trực quan hóa dữ liệu"):
            st.markdown("(*Lưu ý: Trực quan hóa dựa trên những thuộc tính dạng số*)")
            selected_columns = st.multiselect("Chọn cột", options=st.session_state.numerical_columns, key='select_visualize_data')
            st.session_state.selected_columns = selected_columns
            # Kiểm tra xem selected_columns đã được chọn hay chưa
            if not selected_columns:
                # Nếu chưa chọn, gán selected_columns là một cột ngẫu nhiên từ numerical_columns
                selected_columns = [random.choice(st.session_state.numerical_columns)]
            st.session_state.selected_columns = selected_columns

    row1_col1, row1_col2 = st.columns(2)
    # if st.sidebar.button("Xem biểu đồ"):
    if "selected_columns" in st.session_state:
        with row1_col1:
            st.subheader("📈 Biểu đồ đường")
            fig_line = go.Figure()
            # Tạo biểu đồ đường cho từng cột
            for column in st.session_state.df_number[st.session_state.selected_columns]:
                fig_line.add_trace(go.Scatter(x=st.session_state.df_number.index, y=st.session_state.df_number[column], mode='lines', name=column))
            fig_line.update_layout(plot_bgcolor="rgba(0,0,0,0)", paper_bgcolor="rgba(0,0,0,0)")
            st.plotly_chart(fig_line)

        with row1_col2:
            st.subheader("📊 Biểu đồ cột")
            fig_bar = go.Figure()
            # Tạo biểu đồ cột cho từng cột
            for column in st.session_state.df_number[st.session_state.selected_columns]:
                fig_bar.add_trace(go.Bar(x=st.session_state.df_number.index, y=st.session_state.df_number[column], name=column))
            fig_bar.update_layout(plot_bgcolor="rgba(0,0,0,0)", paper_bgcolor="rgba(0,0,0,0)")
            st.plotly_chart(fig_bar)

        st.markdown("-----")

    row2_col1, row2_col2 = st.columns(2)
    with row2_col1:
        st.subheader("📍 Biểu đồ scatter")
        fig_scatter = go.Figure()
        # Tạo biểu đồ scatter cho từng cột
        for column in st.session_state.df_number[st.session_state.selected_columns]:
            fig_scatter.add_trace(go.Scatter(x=st.session_state.df_number.index, y=st.session_state.df_number[column], mode='markers', name=column))
        fig_scatter.update_layout(plot_bgcolor="rgba(0,0,0,0)", paper_bgcolor="rgba(0,0,0,0)")
        st.plotly_chart(fig_scatter)

    st.markdown("-----")

    row3_col1, row3_col2 = st.columns(2)
    with row3_col1:
        st.subheader("heatmap")
        fig_heatmap = go.Figure(data=go.Heatmap(z=st.session_state.df_number.corr().values,
                                                x=selected_columns,
                                                y=selected_columns,
                                                colorscale='Viridis'))
        # Tinh chỉnh layout của biểu đồ
        fig_heatmap.update_layout(
            plot_bgcolor="rgba(0,0,0,0)",
            paper_bgcolor="rgba(0,0,0,0)",
            xaxis=dict(tickmode='array', tickvals=list(range(len(selected_columns))), ticktext=selected_columns),
            yaxis=dict(tickmode='array', tickvals=list(range(len(selected_columns))), ticktext=selected_columns),
        )
        st.plotly_chart(fig_heatmap)

    with row3_col2:
        st.subheader("🥧 Biểu đồ tròn (Pie chart)")
        # Tính tỷ lệ phần trăm của từng cột
        column_percentages = (st.session_state.df_number[st.session_state.selected_columns].sum() / st.session_state.df_number[st.session_state.selected_columns].sum()).sum() * 100
        fig_pie = px.pie(values=column_percentages, names=column_percentages.index)
        st.plotly_chart(fig_pie)

    else:
        st.warning("Không có cột nào chứa dữ liệu dạng số trong dữ liệu.")

```

Hình 4.48: Source code chức năng trực quan hóa dữ liệu.

Phân tích dự báo

Chức năng của phân tích dự báo sẽ bao gồm 3 phần chính, đó là tiền xử lý dữ liệu, huấn luyện mô hình và dự báo dựa trên tập dữ liệu mới. Chức năng của mục tiền xử lý dữ liệu sẽ bao gồm các bước :

Bước 1: Xử lý giá trị khuyết (trên giá trị số).

Bước 2: Loại bỏ ngoại lai (trên giá trị số).

Bước 3: Mã hóa dữ liệu dạng chữ.

Bước 4: Hợp nhất 2 dạng dữ liệu.

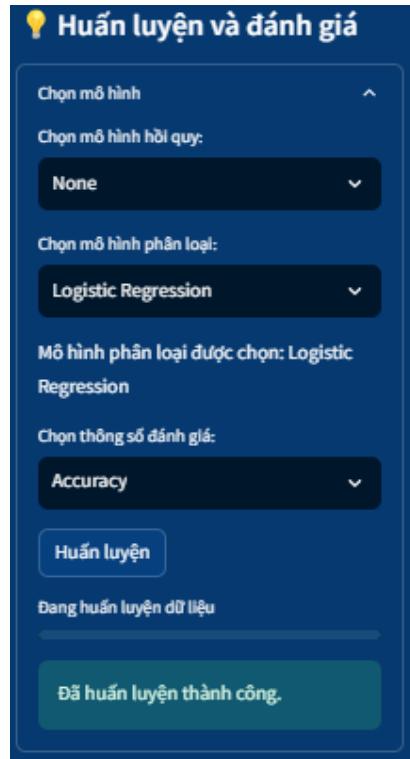
Bước 5: Chia tập dữ liệu X và Y.

Bước 6: Chuẩn hóa dữ liệu.



Hình 4.49: Chức năng tiền xử lý dữ liệu.

Sau khi đã chuẩn hóa dữ liệu thành công, lúc này ta có thể đưa tập dữ liệu vào huấn luyện. Lúc này, người dùng cần nhận định rõ bài toán của mình đang nằm ở bài toán hồi quy hay phân loại để có thể chọn được mô hình phù hợp. Sau khi chọn thành công mô hình, hệ thống sẽ hiển thị lên mục “chọn thông số đánh giá” phù hợp với loại bài toán để người dùng có thể xem được độ chính xác của mô hình vừa chọn.



Hình 4.50: Chức năng huấn luyện mô hình.

```

st.sidebar.title("Huấn luyện và đánh giá")
with st.sidebar.expander("Chọn mô hình"):
    regression_models = ["None", "Linear Regression", "Decision Tree Regression", "SVM", "Ridge"]
    classification_models = ["None", "Logistic Regression", "Naive Bayes", "Support Vector Machine", "Random Forest", "KNN", "Naïve Bayes"]
    st.session_state.selected_regression_model = st.selectbox("Chọn mô hình hồi quy", options=regression_models, index=0)
    st.session_state.selected_classification_model = st.selectbox("Chọn mô hình phân loại", options=classification_models, index=0)
if st.session_state.selected_regression_model == "None" and st.session_state.selected_classification_model == "None":
    st.warning("Vui lòng chỉ chọn một mô hình hồi quy hoặc một mô hình phân loại.")
else:
    if st.session_state.selected_regression_model != "None":
        st.session_state.model, st.session_state.V_train, st.session_state.V_test, st.session_state.selected_regression_model
        st.session_state.model, st.session_state.V_train, st.session_state.V_test, st.session_state.selected_regression_model, st.session_state.X_train_scaled, st.session_state.X_test_scaled, st.session_state.Y_train, st.session_state.Y_test
        if st.session_state.selected_regression_metric != "None" and st.session_state.selected_regression_metric in st.session_state.V_train.keys():
            st.session_state.selected_metric = st.selectbox("Chọn thông số đánh giá", options=list(st.session_state.regression_metrics.keys()))
        if st.session_state.selected_regression_metric == "None":
            if st.button("Đưa vào"):
                progress_bar = st.progress(0)
                for percent_complete in range(100):
                    if (percent_complete * 100) % 10 == 0:
                        st.write(f"Đang huấn luyện dữ liệu ({percent_complete}%)")
                    progress_bar.progress(percent_complete)
                st.success("Đã huấn luyện thành công.")
            else:
                st.warning("Vui lòng chọn một mô hình hồi quy")
    elif st.session_state.selected_classification_model != "None":
        st.session_state.model, st.session_state.V_train, st.session_state.V_test, st.session_state.selected_classification_model
        st.session_state.model, st.session_state.V_train, st.session_state.V_test, st.session_state.selected_classification_model, st.session_state.X_train_scaled, st.session_state.X_test_scaled, st.session_state.Y_train, st.session_state.Y_test
        if st.session_state.selected_classification_metric != "None" and st.session_state.selected_classification_metric in st.session_state.V_train.keys():
            st.session_state.selected_metric = st.selectbox("Chọn thông số đánh giá", options=list(st.session_state.classification_metrics.keys()))
        if st.session_state.selected_classification_metric == "None":
            if st.button("Đưa vào"):
                progress_bar = st.progress(0)
                for percent_complete in range(100):
                    if (percent_complete * 100) % 10 == 0:
                        st.write(f"Đang huấn luyện dữ liệu ({percent_complete}%)")
                    progress_bar.progress(percent_complete)
                st.success("Đã huấn luyện thành công.")
            else:
                st.warning("Vui lòng chọn một mô hình phân loại")
    if "rate" in st.session_state:
        st.subheader("Thống số đánh giá của (" + st.session_state.selected_metric + ") (" + st.session_state.rate + ")")
        # st.write(st.session_state.rate)

```

Hình 4.51: Source code chức năng huấn luyện mô hình.

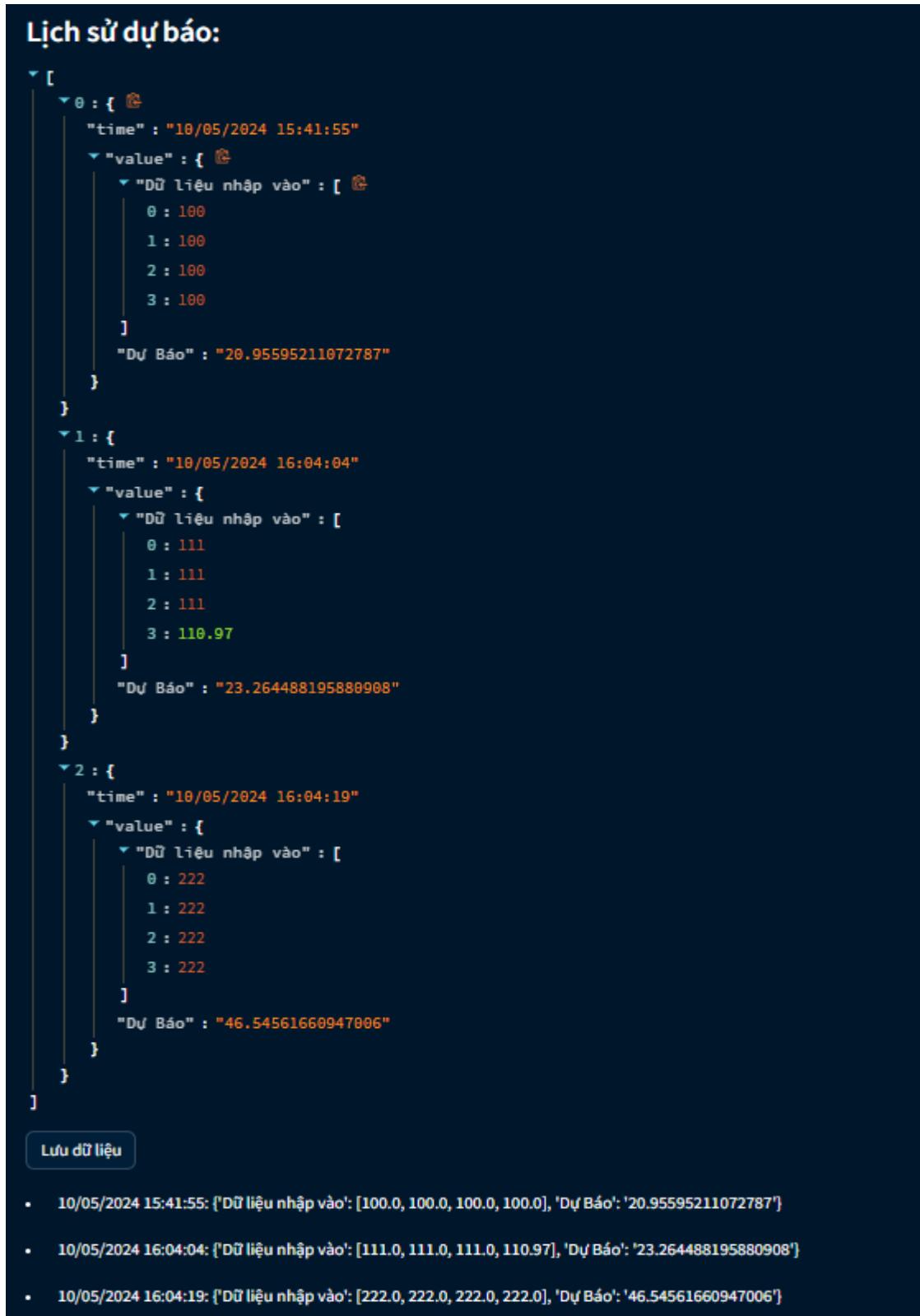
Sau khi đã huấn luyện mô hình thành công, ta có thể tiến hành thực hiện dự báo trên tập dữ liệu mới. Lúc này, hệ thống sẽ hiển thị các ô text tương ứng với các cột dữ liệu đã chọn trong tập X trước đó để người dùng nhập dữ liệu và dự báo.



Hình 4.52: Dự báo trên tập dữ liệu mới.

Lịch sử dự báo

Lịch sử dự báo là chức năng hệ thống lưu lại lịch sử qua các lần dự báo trong 1 session. Dữ liệu dự báo sẽ được hiển thị với điều kiện có các kết quả dự báo. Người dùng có thể xem lịch sử này, thậm chí lưu nó dưới dạng file json để có được 1 tập dữ liệu mới có giá trị hơn.



Hình 4.53: Lịch sử dự báo dữ liệu.

```
{ } predictions.json x  
DATN > { } predictions.json > {} 1  
1 [  
2 {  
3   "time": "09/05/2024 11:44:45",  
4   "value": {  
5     "Dữ liệu nhập vào": [  
6       99.0,  
7       99.0,  
8       99.0,  
9       99.0  
10    ],  
11    "Dự Báo": "97.30268027499312"  
12  }  
13 },  
14 {  
15   "time": "09/05/2024 11:44:54",  
16   "value": {  
17     "Dữ liệu nhập vào": [  
18       111.0,  
19       111.0,  
20       111.0,  
21       111.0  
22     ],  
23     "Dự Báo": "109.4226070105076"  
24   }  
25 },  
26 {  
27   "time": "09/05/2024 11:45:02",  
28   "value": {  
29     "Dữ liệu nhập vào": [  
30       222.0,  
31       222.0,  
32       222.0,  
33       222.0  
34     ],  
35     "Dự Báo": "221.5319293140163"  
36   }  
37 }  
38 ]
```

Hình 4.54: Dữ liệu sau khi được lưu về máy.

KẾT LUẬN

Thời gian làm đồ án tốt nghiệp vừa qua đối với em là thời gian vô cùng thú vị. Em đã tìm hiểu được các kỹ thuật khác nhau của trí tuệ nhân tạo để có thể giải quyết bài toán phân tích dữ liệu. Ngoài ra, em còn có thể vận dụng được các kiến thức đã tích lũy qua quá trình học để có thể tạo nên được một sản phẩm giúp người dùng có thể thực hiện dễ dàng việc phân tích dữ liệu. Em đã học được kỹ năng làm việc độc lập, quản lý thời gian một cách hiệu quả, qua đó nâng cao khả năng học tập của em và sẽ là hành trang vững chắc giúp em bước ra ngoài môi trường giảng đường.

Em đã tìm hiểu nghiên cứu về dự án này và đây cũng là một sự sáng tạo của em khi đã xây dựng 1 sản phẩm của riêng mình, không trùng lặp ý tưởng nào trước đó và em đã tạo ra được cái riêng của riêng mình. Trong quá trình làm dự án, em đã học được rất nhiều thứ và cách thức nghiên cứu cũng như triển khai code của em cũng mạch lạc và rõ ràng hơn. Chúng em cảm ơn thầy Nguyễn Mạnh Cường đã đồng hành cùng em trong quá trình xây dựng đồ án, khiến kiến thức và sự sáng tạo được đi đúng hướng và đúng tiêu chuẩn đầu ra.

Hệ thống đã thành công khi đã phân tích và dự báo được trên các bộ dữ liệu cơ bản thuộc bài toán phân loại và hồi quy. Tuy vậy, do thời gian có hạn và tài nguyên hạn chế, nên hệ thống chưa được toàn vẹn khi chưa có giao diện đẹp, chưa thể thực hiện tốt trên các dữ liệu phức tạp, các chức năng còn hạn chế. Trong tương lai, em sẽ cố gắng phát triển hệ thống thêm nhiều tính năng và có thể đem vào áp dụng trong thị trường.

Em xin được gửi lời cảm ơn chân thành nhất tới thầy giáo, tiến sĩ Nguyễn Mạnh Cường đã tận tình hướng dẫn em thực hiện đề tài này. Em xin chúc thầy luôn luôn mạnh khỏe và thành công trong những nghiên cứu sắp tới.

TÀI LIỆU THAM KHẢO

- [1]. Hồi quy tuyến tính: <https://aws.amazon.com/vi/what-is/linear-regression/>
- [2]. Tìm hiểu mạng nơron: <https://luanvan.co/luan-van/de-tai-tim-hieu-ve-mang-noron-kohonen-17394/>
- [3]. Tìm hiểu về thuật toán cây quyết định:
<https://www.123tailieufree.com/2016/01/su-dung-cay-quyet-dinh-de-phan-loai-du-lieu-nhieu.html>
- [4]. Tìm hiểu về máy vecto hỗ trợ: <https://123docz.net/trich-doan/1300331-bo-phan-loai-vector-ho-tro-support-vector-machine-svm.htm>
- [5] Thiết kế giao diện: <https://www.youtube.com/watch?v=pWxDxhWXJos>
<https://www.youtube.com/watch?v=7yAw1nPareM>
- [6]. Tìm hiểu về streamlit: <https://docs.streamlit.io/>.
- [7]. Hồi quy Ridge: https://phamdinhkhanh.github.io/deepai-book/ch_ml/RidgedRegression.html
https://phamdinhkhanh.github.io/deepai-book/ch_ml/index_RidgedRegression.html
<https://phamdinhkhanh.github.io/2020/08/13/ModelMetric.html>
- [8]. Naive Bayes: <https://solieu.vip/ung-dung-thuat-toan-phan-loai-naive-bayes/>